Chapter 1

What is Data Mining?

- Business analytics methods that go beyond simple descriptive analytics methods and simple data visualization
    - Mithilfe von statistischen Methoden und Machine Learning Methoden können Vorhersagen getroffen werden, um Entscheidungsträger in ihrem Entscheidungsprozess zu unterstützen und datengetriebene Entscheidungen im Unternehmen zu ermöglichen
- Statistical and machine-learning methods to support decision making
- Prediction (at the individual level) is typically an important component

Big Data

Big Data is a relative term and there is no consensus definition on when data is actually *Big Data.* However, Big Data can be characterized by the 4 V's:

- The four V's of data:
    - **Volume:** High volume of data -> often in the size of peta bytes, cant be handled by basic systems from the past
    - **Velocity:** High degree of speed at which the data is being produced and processed ->
    - **Variety:** Data comes from a great variety of sources. Requires distincs processing capabilities and specialist algorithms
    - **Veracity:** Veracity refers to the quality of the data that is being analyzed. High veracity data has many observations that are valuable to analyze and that contribute in a meaningful way to the overall results. Low veracity data, on the other hand, contains a high percentage of meaningless data (causing noise).
- ➔ Data that is high volume, high velocity and high variety must be processed with advanced tools (analytics and algorithms) to reveal meaningful information. Because of these characteristics of the data, the knowledge domain that deals with the storage, processing, and analysis of these data sets has been labeled Big Data.

Overview on data mining methods: Supervised / Unsupervised Learning

Supervised Learning:

Supervised learning is a machine learning approach that's defined by its use of labeled datasets. These datasets are designed to train or "supervise" algorithms into classifying data or predicting outcomes accurately. Using labeled inputs and outputs, the model can measure its accuracy and learn over time.

Unsupervised Learning:

Unsupervised learning uses machine learning algorithms to analyze and cluster unlabeled data sets. These algorithms discover hidden patterns in data without the need for human intervention (hence, they are "unsupervised")

Difference:

The main distinction between the two approaches is the use of labeled datasets. To put it simply, supervised learning uses labeled input and output data, while an unsupervised learning algorithm does not.

In supervised learning, the algorithm "learns" from the training dataset by iteratively making predictions on the data and adjusting for the correct answer. While supervised learning models tend to be more accurate than unsupervised learning models, they require upfront human intervention to label the data appropriately. For example, a supervised learning model can predict how long your commute will be based on the time of day, weather conditions and so on. But first, you'll have to train it to know that rainy weather extends the driving time.

Unsupervised learning models, in contrast, work on their own to discover the inherent structure of unlabeled data. Note that they still require some human intervention for validating output variables. For example, an unsupervised learning model can identify that online shoppers often purchase groups of products at the same time. However, a data analyst would need to validate that it makes sense for a recommendation engine to group baby clothes with an order of diapers, applesauce and sippy cups.

CHAPTER 2 – DATA MINING PROCESS

Supervised vs. Unsupervised Learning

Supervised (observable target variable and outcome variable exists)

- Goal: Predict target or outcome variable
- Target value known in **training data**: Data from which the algorithm (model) learns
- Score to data where value is not known
- METHODS:
    - Classification
    - Prediction

➔ We'll know how good the predictions were

Unsupervised (no target and outcome variable)

- Goal: detect patterns, segment data into meaningful patterns (clustering of customers)
- No target variable to predict or classify (no direct information on how good we did)
- METHODS:
    - Association rules
        - Goal: Produce rules that define "What goes with what?"
        - Association rules are used to find correlations and co-occurrences between data sets. They are ideally used to explain patterns in data from seemingly independent information repositories, such as relational databases and transactional databases. The act of using association rules is sometimes referred to as "association rule mining" or "mining associations."
        - Machine learning models analyze past user behavior data for frequent patterns, develop association rules and use those rules to recommend content that a user is likely to engage with, or organize content in a way that is likely to put the most interesting content for a given user first. ." (zB Content recommendation on Netflix / Spotify
    - Data reduction
        - Reducing number of variables (columns), e.g., by using principal components (Hauptkomponentenanalyse)
        - Reducing the number of records, e.g., by forming groups
    - Data exploration
    - Visualization
        - Help by examining patterns & relationship in data
        - Useful for showing results
        - E.g., histograms, scatter plots, bar charts etc.

Sampling

We apply algorithms or models to a sample from a database (to save computing power, since algorithms compute faster with smaller samples)

➔ Rare Event Oversampling:
    - If the event of interest is rare, the model might not pick up the minority class (zB bei Credit Card Fraud)
    - Random sampling might yield to less of the cases we are interested in
    - Solution: **Oversample** rare cases to obtain a balanced training set
        - Need to adjust for oversampling later on

Creating Binary Dummy Variables

In most algorithms we have to create binary dummy variables for each category of the variable

- Number of dummies = number of categories – 1 [minus one to get rid of redundant information and to avoid collinearity]

Code:

```
Library(dummies)
```

Housing.df.test <- dummy.data.frame(housing.ds, sep="."): Generates dummies for the categories of **all** categorical variables in our housing data frame

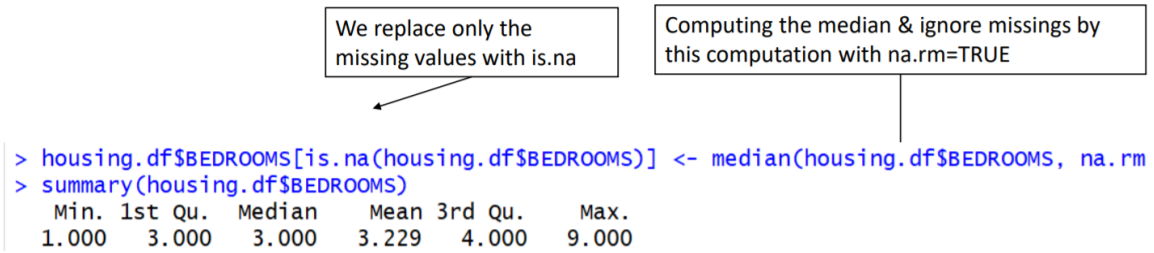
Outliers

The observation that is extreme compared to the rest of the data (is distant)

- These outliers can have a disproportionate influence on our model
- Importance to detect outliers
  - If detected, find out whether its an error or truly extreme (anomaly).
    - Correct if error (e.g., misplaced decimal)
    - If number of outliers is mall (and we find them to be errors): treat as missing value
- How to detect outliers?
  - Graphically (boxplots, scatterplots), order variable (in R, by ordering the table), by looking at minimum/maximum values (zB if minimum value differs a lot from the quartiles or mean/median)

Missing Values

- If for a variable no data value is stored for an observation
- Denoted by NA
- Default in most algorithms is to drop those observations (rows)
  - Dropping them would reduce the number of observations and could lead to skewing
- **Solutions:**
  - Omission: Only practical if number of missings is small (However, this will lead the model to attribute the effect of the missing variables to those that were included, which leads to bias.)
  - Imputation: Replace missings with reasonable substitutes (e.g., mean, median)
    - Advantage: We can keep the record and its non-missing information (other variables of the record)
    - Code:



Normalizing / Standardizing

= adjusting values measured on different scales to a notionally common scale.

- Normalizing puts all variables to the same scale:
  - For example: If a data consists of different units (days, dollars, counts) and we want to use clustering (involves calculating distance measure). When dollar is in 1000s but everything else is in 10s: dollars could dominate the distance measure in clustering
- Formula for Normalizing: Substract the mean and divide by the standard deviation
  - $$z_i = \frac{x_i - \bar{x}}{s}$$ zi: standardized variable, xi: original variable, x_quer: mean of original variable, s: standard deviation of original variable
  - CODE: centered.TAX <- scale(TAX, center=TRUE, scale=TRUE)

- ALTERNATIVE: scale to 0-1 by substracting minimum and divide by range:

$$z_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$

- Formula:
- CODE: install.packages("scales"), library(scales),
  - Rescaled.TAX <- rescale(TAX, to = c(0,1), from = range(TAX, na.rm=TRUE)

Data Partitions & Overfitting

General Goal: find a model that generalizes beyond the dataset we have at hand -> if the data is used is used without partitioning, there is a risk of overfitting. A complex model might have an excellent fit using the data we have **but performs worse on new data.**

- Drawing a line that perfectly fits the points is unlikely to be useful for predicting future prices
- Simple straight line might do a better job

Causes for Overfitting:

- Too many predictors
- Too many parameters in the model (model too complex with too little numbers of observations)
- Trying too many different models (Trying too hard to find the perfect model from the training data -> could lead to overfitting)

The goal is to find the model that does the best job at classifying or predicting. Using the same data to develop (train) and assess the performance of the model, is wrong -> will lead to overfitting

Solution to overfitting:

- Partition the data into:
  - Training set
    - The model is being developed (trained) based on this data set. Estimating coefficients, etc.
    - Typically largest set
  - Validation set
    - Used to assess the performance (Gütemaß, RMSE, adj R^2) of each model and tune parameters. If too many adjustments needed -> Risk of overfitting
    - If we assess multiple models with the same validation data, models could overfit validation data, therefore: a Test set is needed
  - Test set
    - Used to have new data to assess the (fully specified) models performance
    - The test set is a separate set of data used to test the model after completing the training. It provides an unbiased final model performance metric in terms of accuracy, precision, etc
    - Es wird geprüft, wie gut das Modell für unbekannte Daten funktioniert (sowohl in der prediction als auch in der classification)
      - Verhindert Gefahr des Overfittings, die bei Trainings- und Validdatensatz entsteht

Data Partitioning in R

Partitioning into training (50%), validation (30%), test (20%) data sets:

1. Randomly sample 50% of the row IDs for training

   Set.seed(1)
   train.rows <- sample(rownames(housing.df), dim(housing.df)[1]*0.5) <- out of the entire available rows (observations), 50% is sampled randomly
   train.data <- housing.df[train.rows,]

2. Sample 30% of the row IDs into the validation set, drawing only from the records not already in the training set; use setdiff() to find records not already in the training

   valid.rows <- sample(setdiff(rownames(housing.df), train.rows), dim(housing.df)[1]*0.3)
   valid.data <- housing.df[valid.rows,]

3. Assign the remaining 20% row IDs to serve as test set
   test.rows <- setdiff(rownames(housing.df), union(train.rows, vaild.rows)) <- vereinigte Menge aus trainrows und validrows wird abgezogen, nur die restlichen 20% werden partitioniert.
   test.data <- housing.df[test.rows,]

SUMMARY:

Data Mining consists of supervised methods (classification, prediction) and unsupervised methods (association rules, data reduction, data exploration & visualization)

Before algorithms can be appled, data must be explored and pre-proessed

- Checking for outliers and handling them
- Handling missing values
- Generating dummy-variables

To evaluate performance and to avoid overfitting, data partitioning is used

Models are fit to the training partition and assessed on the validation and test partitions

➔ Data Mining methods are sometimes applied to a sample from a large database, and then the best model is used to score the entire database

CHAPTER 3 – DATA VISUALIZATION

Tools zur Visualisierung

Prediction: Outcome is continuous, numeric values (prices, etc)

- Plot outcome on the y-Axis of boxplots, bar charts and scatter-plots
- Study relation of outcome to categorical predictors via side-by-side plots and multiple panels. Study relation of outcome to numerical predictors via scatter plots

Classification: Outcome is binary or categorical (will the customer default / file bankruptcy, etc)

- Study relation of outcome to categorical predictors using bar charts with the outcome on the y-axis
- Study relation of outcome to pairs of numerical predictors via color-coded scatter plots (color denotes the outcome).
- Study relation of outcome to numerical predictors via side-by-side boxplots

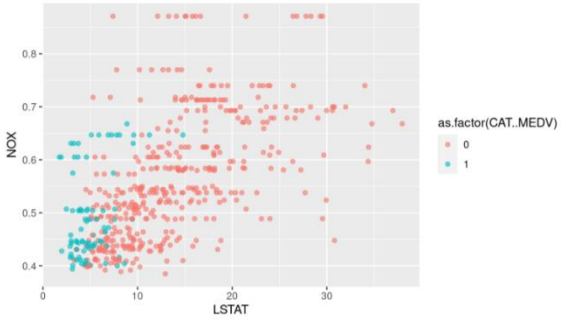
In both cases (prediction and classification)

- Use distribution plots (boxplot, histogram) for determining needed transformations of the outcome variable (and/or numerical predictors)
- Examine scatterplots with added color/panel/size to determine the need for interaction terms.
- Use different aggregation levels for detecting patterns.

Scatterplot

Displays relationship between two numerical variables

- Combined with a colored legend, scatter plots can also display multivariate relationships



Here: multivariate relationship in the case of 3
Cluster: Teure Häuser sind eher in Gegenden mit geringem Prozentsatz von sozioökonomisch schwacher Population und einer niedrigen Nitric Oxide Concentration

Code: ggplot(housing.df, aes(y=NOX, x=LSTAT, color=as.factor(CAT..MEDV))) + geom_point(alpha=0.6)
➔ Alpha beschreibt die Deckungskraft, hilfreich bei sich überschneidenden Werten, hilfreich zur Erkennung von Cluster/Muster/Patterns

Scatter plot matrix

Überblick über die Daten: Scatter Plot matrix offers a combination of bivariate scatter plots, distribution plots and bivariate correlation coefficients

CODE: library(GGally)
ggpairs(housing.df[, c(1, 3, 12, 13)])

Covariance Matrix

Covariance matrix is a square matrix giving the covariance between each pair of elements of a given random vector. Any covariance matrix is symmetric and positive semi-definite and its main diagonal contains variances (i.e., the covariance of each element with itself).

```
> cov(cereals.matrix)
             calories     protein         fat       sodium        fiber       weight         cups       rating
calories  379.6308954  0.40669856  9.77785373  491.0799727 -13.62542720  2.040874915  0.395386193 -188.6815623
protein     0.4066986  1.19856459  0.22966507   -5.0179426   1.30550239  0.035610048 -0.062284689    7.2375603
fat         9.7778537  0.22966507  1.01298701   -0.4562543   0.04010595  0.032505126 -0.041197881   -5.7865386
sodium    491.0799727 -5.01794258 -0.45625427 7027.8537252 -14.12106972  3.892634997  2.334552290 -472.5718370
fiber     -13.6254272  1.30550239  0.04010595  -14.1210697   5.68042379  0.088665243 -0.284567840   19.5575751
weight      2.0408749  0.03561005  0.03250513    3.8926350   0.08866524  0.022643267 -0.006989064   -0.6301718
cups        0.3953862 -0.06228469 -0.04119788    2.3345523  -0.28456784 -0.006989064  0.054156801   -0.6641365
rating   -188.6815623  7.23756028 -5.78653861 -472.5718370  19.55757507 -0.630171813 -0.664136502  197.3263210
```

Correlation Matrix

A correlation matrix is a square matrix showing correlation coefficients between variables. Each cell in the matrix shows the correlation between two variables. A correlation matrix is used to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses. The diagonal (1s) shows that each variable always perfectly correlates with itself.

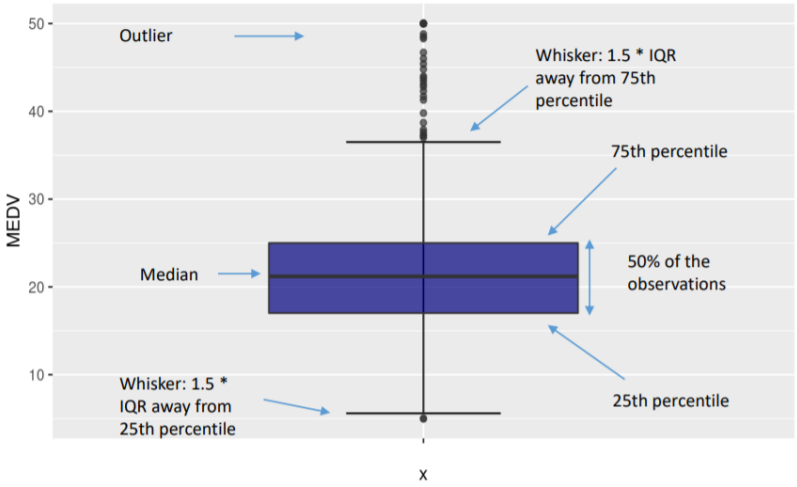
```
> cor(cereals.matrix)
            calories     protein         fat      sodium       fiber     weight       cups      rating
calories  1.00000000  0.01906607  0.49860981  0.30064923 -0.29341275  0.6960911  0.08719955 -0.6893760
protein   0.01906607  1.00000000  0.20843099 -0.05467435  0.50033004  0.2161585 -0.24446916  0.4706185
fat       0.49860981  0.20843099  1.00000000 -0.00540746  0.01671924  0.2146250 -0.17589214 -0.4092837
sodium    0.30064923 -0.05467435 -0.00540746  1.00000000 -0.07067501  0.3085765  0.11966461 -0.4012952
fiber    -0.29341275  0.50033004  0.01671923 -0.07067501  1.00000000  0.2472256 -0.51306093  0.5841604
weight    0.69609108  0.21615849  0.21462503  0.30857645  0.24722563  1.0000000 -0.19958272 -0.2981240
cups      0.08719955 -0.24446916 -0.17589214  0.11966461 -0.51306093 -0.1995827  1.00000000 -0.2031601
rating   -0.68937603  0.47061846 -0.40928366 -0.40129520  0.58416042 -0.2981240 -0.20316006  1.0000000
```

$$\rho = \frac{Cov(x,y)}{\sqrt{Var(x) * Var(y)}}$$

- zB fat and calories positive correlated. If fat increases, calories also increase
- **Correlation heatmap:** Correlation heatmap is graphical representation of correlation matrix representing correlation between different variables.

Boxplot

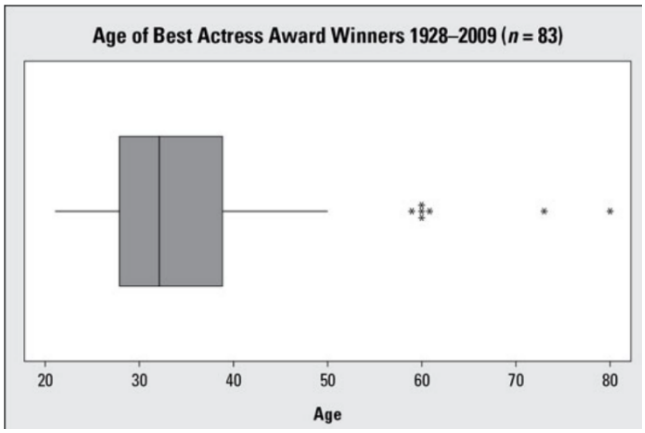
Very useful tool in getting an overview on the overall distribution of a continuous variable. You can't tell the sample size by looking at a boxplot; it's based on percentages of the sample size, not the sample size itself. Although a boxplot can tell you whether a data set is symmetric (when the median is in the center of the box), it can't tell you the shape of the symmetry the way a histogram can. Variability in a data set that is described by the five-number summary is measured by the interquartile range (IQR). The IQR is equal to Q3 – Q1, the difference between the 75th percentile and the 25th percentile (the distance covering the middle 50 percent of the data). The larger the IQR, the more variable the data set is. Notice that the IQR ignores data below the 25th percentile or above the 75th, which may contain outliers that could inflate the measure of variability of the entire data set. So if data is skewed, the IQR is a more appropriate measure of variability than the standard deviation.



Note: IQR = Inter Quartile Range, that is the distance between the 25th and the 75th percentile.

A boxplot can show whether a data set is symmetric (roughly the same on each side when cut down the middle) or skewed (lopsided). A symmetric data set shows the median roughly in the middle of the box.

- Median: is the line that cuts through the box
- Skewness: If the longer part of the box is to the right (or above) the median, the data is said to be skewed right. If the longer part is to the left (or below) the median, the data is skewed left.
- Skewness from descriptive statistics: If the median age (33 years) is lower than the mean age (35.69 years), right skewness is present.



➔ Skewed to the right. The part of the box to the left of the median (representing the younger actresses) is shorter than the part to the right of the median (representing the older actresses). **That means the ages of the younger actresses are closer together than the ages of the older actresses. (lower variance)**

CODE: ggplot(housing.df, aes(x="", y=MEDV)) + stat_boxplot(geom = "errorbar", width = 0.25) + geom_boxplot(width=0.5, fill = "navy", alpha=0.7)

Grouped Boxplot: Allows comparison between categories of a potential predictor

CODE: ggplot(housing.df, aes(x=as.factor(CHAS), y=MEDV)) + stat_boxplot(geom = "errorbar", width = 0.25) + geom_boxplot(width=0.5, fill = "navy", alpha=0.7)

Histogram: Although a boxplot can tell you whether a data set is symmetric (when the median is in the center of the box), it can't tell you the shape of the symmetry the way a histogram can -> Histograms are used for identifying symmetry and the shape of the symmetry.

➔ Wird genutzt um kontinuierliche Variablen darzustellen

CODE: ggplot(housing.df, aes(MEDV)) + geom_histogram(binwidth=5, fill = "navy", alpha=0.8)

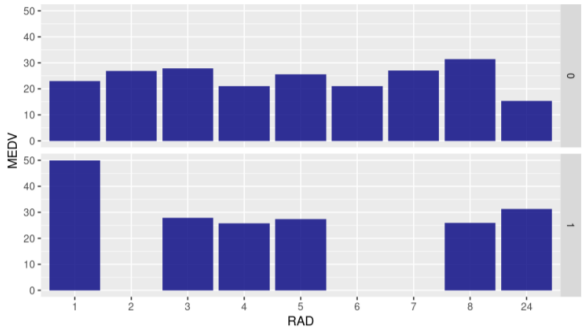
Binwidth= Bandbreite. Gibt an wie Breit die Balken des Histogramms sind.

Alpha= ist die Opazität. 0.5 ist transparent, 1 ist ausgefüllt zb

Bar Chart

Used for:

- Counts of appearances (simple): ggplot(housing.df, aes(x=as.factor(CHAS))) + geom_bar(width=0.5, fill = "navy", alpha=1)
- Depict the mean of two categories of a variable:
    - ggplot(housing.df, aes(x=as.factor(CHAS), y=MEDV)) + geom_bar(width=0.5, fill = "navy", alpha=1, stat = "summary", fun.y = "mean")
- Side-by-side bar chart: used to visualize multivariate relationships. Hier: Beziehung zwischen MEDV und Distanz zu Highways (1,2,3,4,5,6,..,24km) und zusätzlich ob der tract am Charles river liegt (1) oder nicht (0).

- 

- ggplot(housing.df) + geom_bar(aes(x = as.factor(RAD), y = MEDV), stat = "summary", fun.y = "mean", fill = "navy", alpha=0.8) + xlab("RAD") + facet_grid(CHAS ~ .)

Barplot / Bar Charts Rule: Bar charts must have a zero baseline! When our eyes interpret bar charts, we are comparing the relative heights of the bars. When we cut the height off at something greater than zero, it skews this visual comparison, over-emphasizing the difference between the bars in a way that simply isn't honest.

Balkendiagramme müssen eine Null-Basislinie haben! Wenn unsere Augen Balkendiagramme interpretieren, vergleichen wir die relativen Höhen der Balken. Wenn die Basislinien nicht bei Null beginnen, wird dieser visuelle Vergleich verzerrt und der Unterschied zwischen den Balken auf eine Weise überbetont, die einfach nicht ehrlich ist. Mit einer Null-Basislinie kann man den Unterschied zwischen zwei Balken deutlich besser und realistischer einschätzen.

Line graph: Dots are connected, often due to timing. Use case: Time series displaying/analysis. Not suitable for cross section data (Querschnittsdaten). Also wird verwendet, wenn die Punkte miteinander verknüpft sind, oft aufgrund eines zeitlichen Ablaufs. Für Zeitreihenplots geeignet.

ggplot(trains.df,aes(x = as.Date(Month), y = Ridership)) + geom_line()+ geom_point()

Exploring the data

Parallel to visualizing the data, the first step should always be to calculate aggregate summary statistics:

- Mean (average)
- Median
- Minimum
- Maximum
- Standard deviation (eher bei continuous values)
- Counts and percentages (eher bei categorical variables)
- Calculate these summary statistics with: describe(housing.df)

Frequency tables

Table(housing.df$CHAS)

Cross Tabulation

Table(housing.df$CHAS, housing.df$CAT..MEDV)

CHAPTER 4 – PERFORMANCE EVALUATION

Goal: identify model that best predicts new records (validation data)

- Predictive accuracy =|= goodness-of-fit

Leitende Frage / Leading Question Performance Evaluation: Wie gut ist das Modell darin, in neuen Daten (Validierungsdaten und Testdaten) die Outcome Variable vorherzusagen? (Nicht: wie gut passt das Modell zu den Daten, auf denen es trainiert wurde?)

PERFORMANCE EVALUATION FOR CLASSIFICATION

Terminology:

Error: classifying a record as belonging to one class when it belongs to another. Fehler wenn das Model eine falsche Vorhersage trifft, also der falsch klassifiziert.

Error rate: % of misclassified records / total records in the validation data / (der prozentuale Anteil an missklassifizierten Einträgen an allen Einträgen im Datensatz):

- Man trainiert sein Modell auf den Trainingsdaten und macht dann mit diesem modell eine Vorhersage auf dem Validierungsdatensatz. Z. B. wenn Validierungsdatensatz 100 Einträge hat und man diese mit dem Modell aus dem Trainigsdatensatz vorhersagen würde und davon 10 falsch klassifiziert wurden, hat das Modell eine 10% error rate.

Naive rule: classify all records as belonging to the most prevalent class (this will be the benchmark) / Alle Einträge im neuen Datensatz werden als der wahrscheinlichere Ausgang klassifiziert -> Liefert keine besonders gute Vorhersage. Modell muss besser sein als die Naïve Rule

- Alle Einträge im Validierungsdatensatz werden der Gruppe zugeordnet, die am häufigsten vertreten ist
- For example: if the confusion matrix shows that 0 is the most prevalent class, then the naïve rule would be to just classify all records as 0's.

Separation of records: How good the model manages to separate positives from negatives. // Wie gut kann das Modell die Beobachtungen in die einzelnen Klassen einteilen?

High separation of records: Using predictor variables attains low error -> Wenige Fehler in der Klassifizierung

Low separation of records: Using predictor variables does not improve much on naïve rule -> Modell schlägt die Naïve Rule nicht besonders gut

<u>Assess classification</u>

For each record:

1. Compute probability (propensity) of belonging to class "1" according to the developed model (from the training data set)
2. Compare to cutoff value, classify accordingly
    a. If the predicted probability is higher than 0.5, classify the record as "1"

<u>Confusion Matrix</u>

Shows overall error rate, accuracy, which records were correctly classified, false positives, false negatives, etc.

- In certain situations, False negatives can be costly (asymmetric costs) whereas
- It is important to differentiate between False Negatives and False Positives due to asymmetric costs
- Overall Error rate: (False Negatives + False Positives) / Gesamtzahl an Beobachtungen
- Accuracy: 1 – Overall Error Rate oder (True Positives + True Negatives)/Gesamtzahl…

<u>Cutoffs</u>

Sensitivity: True Positive Rate. the proportion of samples that are genuinely positive that give a positive result using the model. The higher the better the model can predict true positive values

Specificity: True Negative Rate. Is the proportion of samples that test negative using the model in question that are genuinely negative. The higher the better the model is at picking up true negative values

Unterscheidung False Negative / False Positive: Unterscheiden sich in den potenziellen Kosten zb. Hohe False Negative Rate bei einem Corona Test ist fataler (kostenintensiver) als eine hohe False Positive Rate

Lowering the cutoff value leads to more positive predictions:

- False Positives ↑ Sensitivity ↑ (since TP/(TP+FN))
- True Positives ↑ Specificity↓ (since TN/(TN+FP))

Increasing the cutoff value leads to more negative predictions

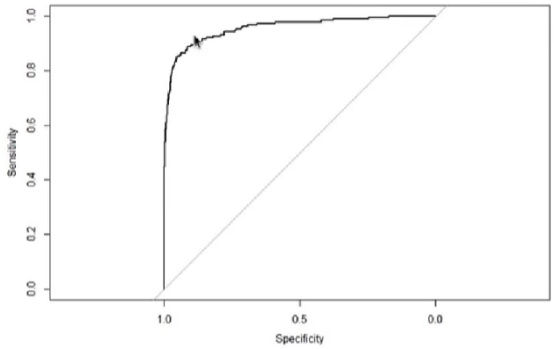
- True Negatives ↑ Sensitivity ↓
- False Negatives ↑ Specificity ↑

Special Case: cutoff value of 1 leads to zero sensitivity and one specificity

Receiver Operating Curve (ROC)

Illustration of sensitivity and specificity as the cutoff value descends from 1 to 0. // Abbildung von Sensitivität und Spezifität bei sinkendem Cutoff Value von 1 bis 0

- Is used to find the optimal cutoff value
- Depicts the trade-off between sensitivity and specificity



Best performance is achieved if the curve is closest to the top-left corner, because then, high sensitivity and high specificity is present.

In this example, the separation of records is high. The closer the ROC is to the Naïve Rule, the lower the separation of records.

Each point of the ROC curve represents a confusion matrix for a specific cutoff value

- ROC helps to understand how well different cutoff values work for one specific model
- AUC is a number for comparing different models

Area under the curve:

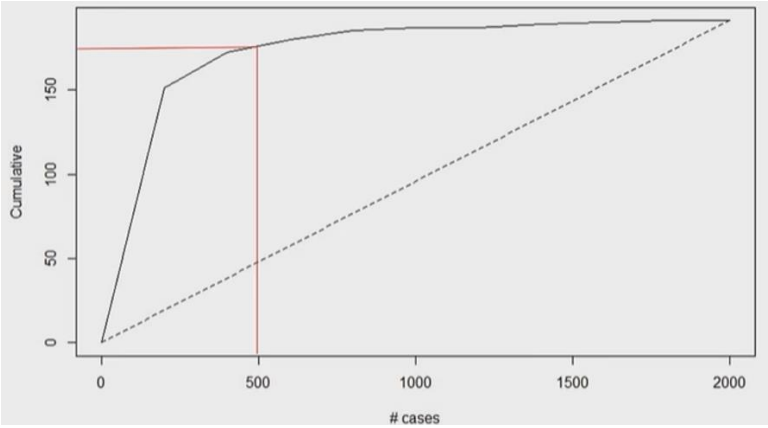
0.5: no better than naïve rule
1.0: perfect separation of classes

-> The higher the AOC the better

Andere Möglichkeit um Performance zu evaluieren: Lift chart and decile-wise lift chart

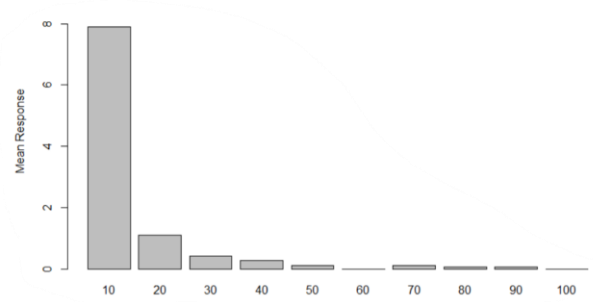
Vorhergesagte Werte werden nach Wahrscheinlichkeit sortiert. Somit wird sichergestellt, dass die wahrscheinlichsten Cases abgedeckt sind -> Wichtig für Kostenoptimierung z.B.

Assessing the performance in terms of identifying the most important class. Using the model's classifications, sort records from most likely to least likely members of the important class



➔ ordered probabilities: For the first 500 predicted records, the model correctly identifies ~170, whereas randomly picking 500 from the dataset would only yield ~50 correct predictions

Decile-wise chart



Taking the 10% of the records that are tanked by the model as the most probable 1's, yields almost 8 times as many 1's as a random 10% selection would.

"mailing to the 10% most promising customers, aka classified as the most probable 1's by the model, will yield 8 times higher response rates than randomly picking 10% of the population to mail"

Asymmetric costs

Situation, in der Kosten für eine Fehlklassifizierung einer Beobachtung größer sind in der einen Klasse als in der anderen. Oder der Benefit einer korrekten Klassifizierung in der einen Klasse größer als in der anderen (z. B. Bankknden die am ehesten ein Darlehen aufnehmen würden)

Cost of making a misclassification error may be higher for one class than the other(s). Or: benefit of correct classification may be higher for one class than the other(s)

zB

- Identify customers that are likely to buy a product (forgone sales, identifying those who are most likely to buy and directing advertisement towards those people is much more efficient than just randomly selecting people to randomly send ads to)
- Identify fraudulent tax declarations

Asymmetrische Kosten können zu Situationen führen, in denen Modelle mit schlechterer Accuracy als der Naive Rule bevorzugt werden, weil sie die interessierende Klasse besser einfangen

Costs and benefits in practice

Sometimes actual costs and benefits are hard to estimate. Lösung: Alles im Sinne der Kosten angeben (Also für jeden Record tatsächliche Kosten und Opportunitätskosten aufstellen -> statt benefits from sale guckt man sich opportunity costs of lost sale an)

- ➔ Führt zu den selben Entscheidungen aber ist oft einfacher durchzuführen, weil diese Kostendaten einfacher zu bekommen sind
- ➔ Ziel ist es hierbei die durchschnittlichen Kosten pro Record zu minimieren

Goal is to minimize average cost per record (Costs for falsely classified 1s and falsely classified 0s)

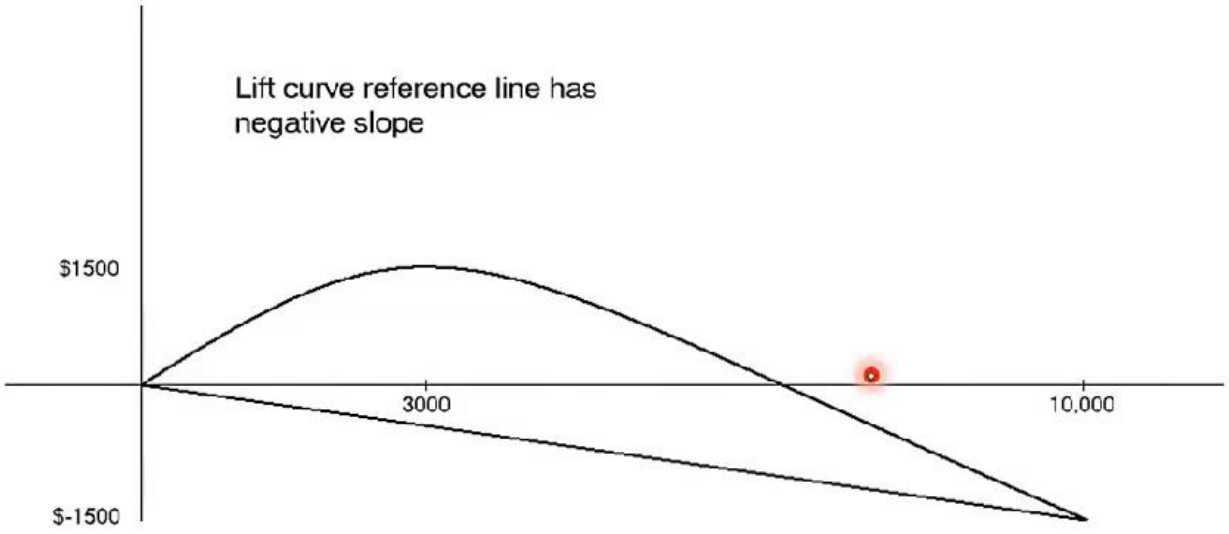
Ratio of misclassification costs:

- good practical substitute for individual costs ("eg misclassifying fraudulent firms is 5 times worse than misclassifying solvent firms)
- q1 = cost of misclassifying an actual "1"
- q0 = cost of misclassifying an actual "0"
- ⇨ Goal: optimize cost ratio q1/q0

Lift chart with costs and benefits

1. Sort records in descending probability of success // Beobachtungen werden nach höchster Wkt geordnet
2. For each case, record cost/benefit of actual outcome // Für jeden Fall werden die Kosten des tatsächlichen Outcomes angegeben
3. Also record cumulative cost/benefit // Kosten werden aufsummiert
4. Plot all records
    a. X-axis is index number (1 for 1st case, n for nth case) // geordnet nach 1. Fall, 2. Fall, 3. Fall etc
    b. Y-axis is cumulative cost/benefit
    c. Reference line from origin to y_n (y_n = total benefit) //Summe aller Kosten/Einnahmen über die Beobachtungen

Plot: Negative slope to reference curve:



Auf der X-Achse sind die Anzahl der Beobachtungen, nach der Höhe der Wahrscheinlichkeit für einen Treffer (1) geordnet (also Beobachtungen nah am Ursprung haben die höchste Wahrscheinlichkeit True Positive zu sein). Die Gewinne die durch diese Ordnung erzielt werden, werden aufsummiert. Wenn man die ersten 3000 nimmt, würde man also beim höchsten Gewinn von $1500 landen. Somit sollte man die ersten 3000 nehmen. Hier ist Kosten/Nutzen maximiert. Reference Line verläuft unterhalb der X-Achse. Reference Line hat negative Steigung, weil total net benefit aller Fälle zusammen negativ ist.

<u>Oversampling</u>

Asymmetric costs/benefits typically go hand in hand with presence of rare but important classes.

Wenn man das Problem hat, dass die Klassen für die man sich interessiert, besonders wenig repräsentiert sind im Datensatz. Oversampling heißt, dass man die Anzahl der 1en künstlich erhöht, um die Anzahl der seltenen Fälle (für die man sich interessiert) im Datensatz überzugewichten

- Customers accepting a loan
- Responder to mailing
- Someone who commits fraud
- Debt defaulter

➔ Often we oversample rare cases to give model more information to work with. Typically use 50% “1” and 50% “0” for training data set.

➔ Wichtig: man “bläht” die Anzahl der 1en im Trainingsdatensatz auf, aber nicht im Validierungsdatensatz. Im Valid.datensatz hat man wieder das originale Verhältnis zwischen 1en und 0en, um zu sehen wie das Modell auf realen Daten performen würde

Oversampling can also be used to model asymmetric costs:

die 0s werden (aufgrund der 5-fach teureren Kosten) nun mit dem Faktor 5 in den Trainingsdatensatz aufgenommen, um zu repräsentieren, dass die Fehlklassifizierung 5 mal teurer ist.

Hier entstehen zwar 2 Fehlklassifizierungen von den X'en, jedoch hat man mit diesem Cut-Off value auch die Fehlklassifierung der O's verhindert, war andernfalls sehr teuer (5 mal so teuer wie Fehlklassifizierung von X) gewesen wäre.

- ➔ Die Frage wie viel man oversampled hängt davon ab wie stark das Verhältnis der asymmetrischen Kosten ist

<u>Oversampling procedure zusammengefasst</u>

1. Separate the responders (rare) from non-responders // 1er und 0er vom Datensatz trennen
2. Randomly assign half the responders to the training sample, plus equal number of non-responsers -> Dadurch 50:50 Trainingsdatensatz aus 1ern und 0ern
3. Remaining responders go to validation sample // Rest der 1er fließen in den Valid.datensatz
4. Add non-responders to validation data, to maintain original ratio of responders to non-responders (Der Validierungsdatensatz wird Solange mit 0ern aufgefüllt, bis das originale Ratio aus 1ern und 0ern aus dem Grunddatensatz wiederhergestellt ist. Validierungsdatensatz wird absolut gemessen dann nur noch halb so gross sein wie der ursprüngliche Datensatz) -> Jetzt auf Trainingsdatensatz trainieren und kalibrieren und auf dem Valid.datensatz die Performance testen.
5. Wenn Test-Datensatz benötigt wird, Randomly take test set (if needed) from validation (bevor man auf dem Validierungsdatensatz testet)

<u>Classification using triage</u>

- ➔ Graue Zone für cut-off value einführen. Modell wird so eingestellt, dass z.B. alle Werte über 0.6 als 1 klassifiziert werden und alle Werte unter 0.4 werden als 0 klassifiziert. Die Werte zwischen 0.6 und 0.4 werden dann manuell überprüft und klassifiziert. Wichtig z.B. in der Medizin bei CT-Scans

Take into account a gray area in making classification decisions.

Example: every predicted record above 0.8 gets classified as 1, and every predicted record below 0.2 gets classified 0. But those predictions between 0.2 and 0.8 are deemed a third group that might receive special human review where it will be decided manually if the record should be classified as 1 or 0.

z.B. im Medizinbereich: Tumorerkennung

PERFORMANCE EVALUATION FOR PREDICTION

Accuracy Measures (Genauigkeitsmaß): Zum Messen der Fehler die das Modell macht

- (MAE) Mean absolute error: gives an idea of the magnitude of errors: Absolute Mittlere Stärke des Fehlers, positive und negative Fehler heben sich nicht gegenseitig auf
- (ME) Mean error: Hier heben sich positive und negative Fehler gegenseitig auf. Daraus kann man schließen ob man eine systematisch über- oder unterschätzung hat. Wenn ME negative, hat man Überschätzung der Werte
- (MPE) Mean percentage error
- Mean absolute percentage error MAPE
- (RMSE) Root mean squared error: positive und negative werte heben sich nicht auf, aufgrund der Quadrierung. Dient dazu das Ausmaß des Fehlers zu messen, ähnlich wie mean absolute error

Wenn Fehler vollständig symmetrisch, ist Mean Error = 0

Prediction accuracy measures: Code für ME, RMSE, MAE, MPE, MAPE des Validierungsdatensatzes

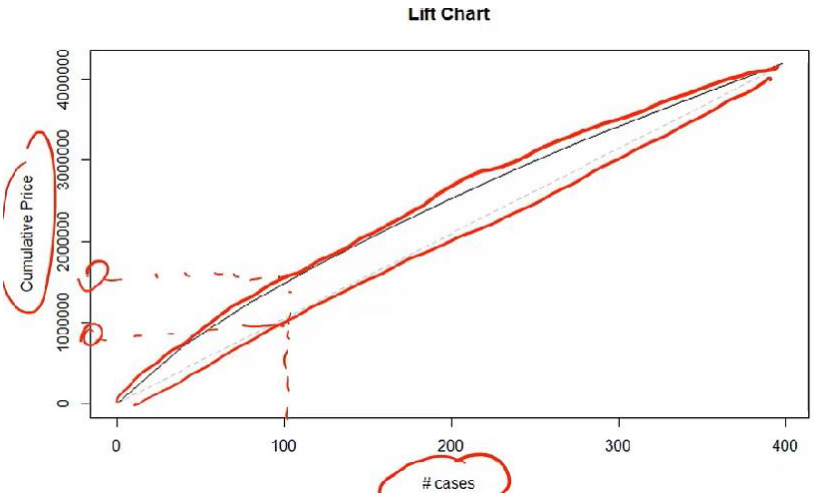
accuracy(pred_v, Toyota.corolla.df[validation,]$Price)

Prediction Lift Chart

Order prediction from highest to lowest (according to interest: e.g., selling price)

Die vorhergesagten Verkaufspreise von groß nach klein ordnen, auf Y-achse der kumulative Preis. Reference Line ist aufsummierte Anzahl der Fälle multipliziert mit dem Mittelwert (Naïve Rule. Die NR bei Prediction Modellen ist einfach der Mittelwert). Der lift chart sollte besser sein als die reference line.

Lift Chart Kurve: Ist der kumulative Preis, den man erzielt, wenn man die Autos in der Reihenfolge verkaufen würde, in der man die vorhergesagten Preise eingeordnet hat. Sollte über der Reference Line liegen, sonst ist das Modell schlecht.



➔ Reference Line ist die aufsummierte Anzahl der Fälle multipliziert mit Mittelwert. Am Punkt 100 (#100 cases) ist der Wert der Reference Line z.B. 100*Mittelwert

➔ "Lift" ist die Verbesserung des Modells gegenüber der Naïve Rule

Wenn man die Top 100 Autos verkauft, die vom Modell geschätzt (und absteigend geordnet) wurden, erhält man einen höheren kumulativen Preis als wenn man random 100 Autos picken würde (Der Lift ist also die Verbesserung gegenüber der Naïve Rule, die das Modell zustande bringt). Selbes gilt für decile-wise Lift chart

CHAPTER 5 – MULTIPLE LINEAR REGRESSION

MLR: lineare Abbildung der Beziehung zwischen einer und mehreren Variablen

X:= Prädiktoren, bzw. Unabhängige Variablen / erklärende Variablen

Scatterplot und regression zeichnen:

```
ggplot(car.df, aes(y=Price, x=HP)) +  geom_point() + expand_limits(x = 0, y= 0) -> scatterplot
                                    +  stat_smooth(method = 'lm', se=FALSE) -> regressionsgerade
```

Selected.var <- erstellt vector über den dann nur bestimmte Variablen (Spalten) betrachtet werden

expand_limit:= setzt den Beginn der X- und Y-Achse jeweils auf Null

How are the betas computed?

Grundidee:

Nur y und X ist bekannt, wie wird Beta geschätzt/bestimmt?

1. Gegeben sei lineare Funktion:
   Let $\tilde{y}_i(b) = bx_i$ with b being a number.
2. Da man b und die Daten hat, kann man nun Residuen berechnen:
   a. $e_i(b) = y_i - \tilde{y}_i(b) = y_i - bx_i$
3. Nun wird die Summe der quadrierten Residuen minimiert. Hierfür wird das "b" errechnet, welches die Summe der quadrierten Residuen minimiert. Dieses "b" ist der optimale Koeffizient Beta_dach
4. OLS versucht also, eine Gerade durch die Datenwolke zu ziehen, die so nah wie möglich an jedem einzelnen Punkt ist.
5. (Quadrieren, damit man den absoluten Fehler berechnet und sich die positive und negative Fehler nicht gegenseitig auscanceln, was das Ausmaß des Fehlers verzerren würde. Außerdem werden größere Abweichungen stärker gewichtet [bzw bestrafen], was das Modell genauer macht)

For computing beta, we need to calculate the residuals first, i.e. yi – yi_hat. There is only one value of beta that minimizes the sum of squared residuals. That particular beta is our least squares estimate.

- ➔ Least squares "tries" to fit a straight line "as close as possible" to the data points.
- ➔ Durch den kleinste quadrate Schätzer wird eine Gerade erzeugt, die den geringsten Abstand zu jedem Punkt in der Datenwolke hat.

Wenn Fehler negativ ist, überschätzen wir

Signifikanzniveau / significance level := Irrtumswahrscheinlichkeit. Gibt an, ob der Zusammenhang zwischen zwei variablen zufällig ist oder tatsächlich besteht. P-Wert sagt aus, wie wahrscheinlich es ist, dass der geschätzte Koeffizient von 77.35 stimmt (oder ist dieser Wert in Wirklichkeit Null?)

- ➔ Nullhypothese: Beta ist gleich Null
- ➔ Alternativhypothese: Beta ist ungleich Null (also =77.35)
- ➔ Wenn p-Wert größer als das Signifikanzniveau ist, dann kann man die Nullhypothese nicht ablehnen, d.h. Beta ist nicht significant von Null verschieden
- ➔ Wenn p-wert Kleiner als das Signifikanzniveau ist, kann man die Nullhypothese ablehnen und es besteht Evidenz dafür, dass das Beta ungleich Null ist und tatsächlich ein statistischer Zusammenhang zwischen den beiden Variablen Preis und HP besteht bzw. Und dass dieser Effekt nicht einfach zufällig entstanden ist.
- ➔ Wenn p-Wert Kleiner als 0.05, kann man sagen, dass der Koeffizient significant auf dem 5%-Niveau. Je Kleiner der p-Wert, desto stärker die Evidenz dafür dass der Koeffizient significant ist.

```
Coefficients:
             Estimate Std. Error t value           Pr(>|t|)
(Intercept)  3875.79    1020.21     3.8            0.00016 ***
HP             77.35       9.91     7.8 0.000000000000027 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- ➔ Wie wahrscheinlich ist es, dass man wirklich ein Beta von 77.35 beobachtet? Ist es in WIrklichkeit nicht doch vielleicht 0? Die Nullhypothese ist hier, dass das Beta gleich Null ist. Die Alternativhypothese ist, dass das Beta ungleich 0 (=77.35) ist.
- ➔ Wenn P-Wert größer als das postulierte Signifikanzniveau (zB 1%, **), dann kann man die Nullhypothese bei einem Signifikanzniveau von 1% nicht ablehnen (d.h. Beta = 0).
- ➔ Wenn P-Wert Kleiner als das Signifikanzniveau (hier sogar Signifikanzniveau von 0.1% -> 99.9% Konfidenzintervall), dann lehnt man die Nullhypothese ab. Es gibt also Evidenz für einen Zusammenhang zwischen HP und dem Preis. Das beta ist also von Null verschieden. HP ist also sehr significant auf dem 0,1% Niveau)

(EXKURS R: options(scipen = 999) reduziert die Anzahl der Nachkommastellen)

Multiple Lineare Regression

Car.lm <- lm(Price ~., data = train.df) #speichert ein Objekt names car.lm in dem der Preis auf alle anderen Daten aus dem Datensatz regressiert wird.

```
                Estimate  Std. Error t value             Pr(>|t|)
(Intercept)  -1774.87783  1643.74482   -1.08                0.281
Age_08_04     -135.43088     4.87591  -27.78 < 0.0000000000000002 ***
KM              -0.01900     0.00234   -8.12  0.0000000000000028 ***
Fuel_TypeDiesel 1208.33916   534.43140    2.26                0.024 *
Fuel_TypePetrol 2425.87671   520.58798    4.66  0.0000039169767967 ***
```

- ➔ Beim kategorisieren wird eine Klasse rausgelassen (hier CNG), um Multikolleanirität zu vermeiden. Aussagen über Diesel und Petrol sind hierbei immer im Vergleich zu CNG zu treffen (Basiskategorie)
- ➔ Wenn der Wert eines Koeffizienten zu einer Variable sinkt, je mehr erklärende Variablen man ins Modell aufnimmt, lässt dies vermuten, dass man vorher im einfachereren Modell (mit weniger erklärenden Variablen) den Einfluss jener Variable überschätzt hat.

Wenn das Auto ein Benziner ist, steigt der Preis im Durchschnitt um 2425.9€ im Vergleich zum CNG-Auto (ausgelassene Referenzklasse)

- Wenn sich Koeffizienten von simplen Modellen stark von denselben Koeffizienten in komplexeren Modellen unterscheiden, bedeutet dies, dass die Koeffizienten im simplen Modell korrelieren und verzerrt sind. Beim hinzufügen von mehreren Variablen (im kompexeren Modell) kann man einen Koeffizienten besser isolieren und einen wahrheitsgetreueren Wert schätzen.

Bivariate Estimation:

```
car.lm <- lm(Price ~ ., data = train.df) # here we use the training dataset.
# use options() to ensure numbers are not displayed in scientific notation.
options(scipen = 999)
summary(car.lm)

                  Estimate  Std. Error t value             Pr(>|t|)
(Intercept)     -1774.87783  1643.74482   -1.08                0.281
Age_08_04        -135.43088     4.87591  -27.78 < 0.0000000000000002 ***
KM                 -0.01900     0.00234   -8.12   0.0000000000000028 ***
Fuel_TypeDiesel  1208.33916   534.43140    2.26                0.024 *
Fuel_TypePetrol  2425.87671   520.58798    4.66   0.0000039169767967 ***
HP                 38.98554     5.58718    6.98   0.0000000000081162 ***
Met_Color          84.79272   126.88345    0.67                0.504
Automatic         306.68415   289.43314    1.06                0.290
CC                  0.03197     0.09908    0.32                0.747
Doors             -44.15774    64.05653   -0.69                0.491
Quarterly_Tax      16.67734     2.60267    6.41   0.0000000003028702 ***
Weight             14.66749     1.53659    8.24   0.0000000000000011 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

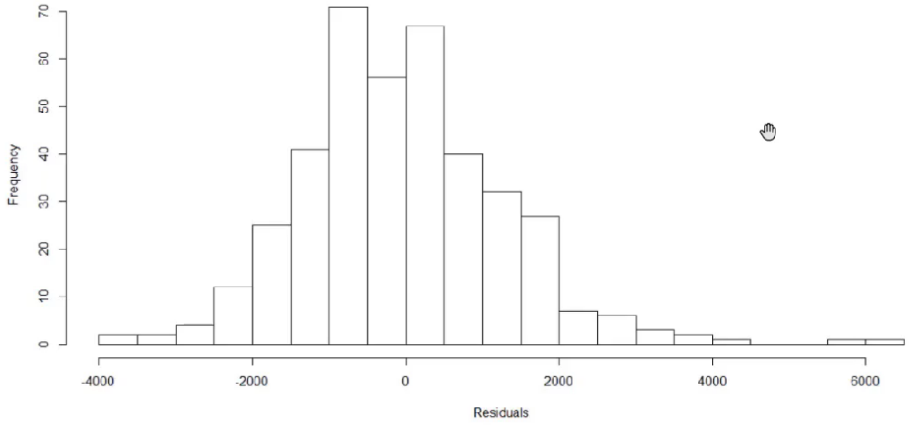
- ➔ "Keeping all others constant, a one unit increase in HP leads to a 38.9€ increase in price on average"

Warum sinkt der RMSE wenn man mehr Variablen in das Modell aufnimmt? / Why does the RMSE decrease when increasing the number of variables in the model?

- ➔ Mit der kleinste Quadrate Methode minimiert man die quadrierten Fehler: Je mehr Variablen man aufnimmt, desto mehr kann man von der Variation der abhängigen Variable (y) erklären. ABER: wenn man zu viele Variablen aufnimmt besteht das Gefahr der Überanpassung/Overfitting.

➔ Indikator für Overfitting/Überanpassung: Histogramm der Residuen: Fehler (residue) sind stochastisch und können nicht vorhergesagt werden durch das Modell. Wenn man jedoch ein Muster in den Fehlern erkennen kann, lässt dies darauf schließen, dass in dem Modell noch irgendeine Variable fehlt die man nicht eingefügt hat oder über die man keine Daten hat. D.h. dass die x-variablen sich auch untereinander beeinflussen (korrelieren), was nicht sein sollte.

➔ Wenn man also ein Muster in den Residuen sieht, lässt dies darauf vermuten, dass im Model noch irgendeine Variable fehlt, die man entweder nicht eingefügt hat, oder über die man keine Daten hat. Oder aber es lässt darauf schließen, dass Prädiktoren sich untereinander beeinflussen, was nicht sein sollte, da man einen Effekt bzw Variable isoliert betrachten will



- Symmetric
- Few outliers

- Hier sind die Fehler relativ normalverteilt. Man erkennt kein Muster, das großartig von der standardnormalverteilung abweicht, erkennt man nicht. Also ist das Modell nicht overfitted

Woher weiß man, welche Variablen man ins Modell aufnehmen sollte?

1. Exhaustive Search / Erschöpfende Suche / Bruteforce Methode

Probiert alle möglichen Kombinationen von den X-Variablen und schaut, wie sie performen (mithilfe des adjustierten R^2). Für jede Anzahl an beliebigen X-Variablen kann dadurch die optimale Kombination identifiziert werden.

- R^2 / coefficient of determination:
  - Formel: ESS / TSS: (Summe der quadrierten Residuen) / (Totale Quadratsumme)

$$R^2 = \frac{ESS}{TSS} = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

  - Erklärt, wie viel der Variation in der abhängigen Variable durch das Modell erklärt wird
  - adjR^2 = 0.3 heißt, dass 30% der Streuung der abhängigen Variable Y wird durch das lineare Modell erklärt.

- R^2 Steigt mit steigender Anzahl an Variablen -> Gefahr der Überanpassung -> Adjustiertes R^2 -> Bestraft zu viele Variablen

$$R^2_{adj} = 1 - \frac{n-1}{n-p-1}(1 - R^2)$$

N= Anzahl
P=ANzahl unabhängiger Variablen/Prädiktoren

- Code Exhaustive Search:

```
search <- regsubsets(Price ~ ., data = train.df, nbest = 1,
       nvmax = dim(train.df)[2], method = "exhaustive")
```

Nvmax gibt maximale Anzahl der Variablen im Model an. Dim [2] bezieht sich auf die Anzahl der Spalten (Variablen)
Hierbei fallen jedoch zwei Variablen raus: Einmal die abhängige Variable (y) und (in diesem Fall) die dritte Klasse der Faktorvariable, um Multikollinearität zu verhindern
Ergebnis regsubsets:

  - CP:= Mallows' CP -> Gütemaß ähnlich wie adjR^2 -> Je Kleiner desto besser
  - Which zeigt die verschiedenen modelle an
  - Rsq:= R^2
  - Adjr2:= adjustiertes R^2

- Adjustiertes R^2: Das R^2 ist ein Gütemaß der linearen Regression. Es lässt sich interpretieren als der Anteil der Varianz der abhängigen Variable, der durch die unabhängigen Variablen erklärt werden kann. Da dieser Wert jedoch mit steigender Anzahl an unabhängigen Variablen natürlicherweise steigt, ist er nur begrenzt aussagekräftig. Hilfe schafft hier der adjusted R^2, der zusätzlich zum R^2 einen Strafterm einfügt. Die "Strafe" steigt hierbei mit steigender Anzahl an unabhängigen Variablen. Somit ist das adj R^2 natürlicherweise Kleiner als das standard R^2. Somit gilt: Durch Hinzunahme einer neuen Variablen kann das Modell im Sinne des korrigierten R^2 nur dann verbessert werden, wenn der zusätzliche Erklärungsgehalt den Strafterm mehr als ausgleicht.

<u>Forward Selection:</u>

Hier wird komplett ohne Prädiktoren gestartet. Dann wird nacheinander immer ein einzelner Prädiktor eingefügt und nach jedem Einfügen wird das adj. R^2 geprüft, ob es sinkt oder steigt. Die Variable, die das adj. R^2 am meisten erhöht, wird aufgenommen. Wenn das nächste Einfügen nicht mehr zu einem signifikanten Anstieg des adj. R^2 führt, hört dieser Algorithmus auf.

- Vorteile (im Vergleich zu exhaustive)
  - Weniger Rechenleistung erforderlich (interessant aus Kostengründen)

<u>Backward Elimination</u>

Hier werden alle Prädiktoren (Variablen) aufgenommen. Dann werden sukzessiv (nach und nach) die Variablen rausgeschmissen, die Prädiktoren rausgeschmissen, die am wenigsten (nicht significant) zum adj. R^2 beitragen

⇨ Für alle drei Methoden gilt: Das Entfernen von überflüssigen Prädiktoren ist der Schlüssel zu einem genauen und robusten Modell. Durch jeder dieser drei Methoden können redundante Prädiktoren ausfindig gemacht und entfernt werden

Koeffizienten interpretieren

Model	Name	Interpretation	Meaning	Example
$y = \beta x + \varepsilon$	Level-level regression	$\hat{\beta} = \Delta y \mid_{\Delta x=1}$	If x increases by one unit, then y changes by $\hat{\beta}$ units.	$x = HP$ $y = price$
$\log y = \beta x + \varepsilon$	Log-level regression	$\hat{\beta} = \frac{\Delta y}{y} \mid_{\Delta x=1}$	If x increases by one unit, then y changes by $\hat{\beta} \cdot 100$ percent.	$x = education$ $y = wage$
$y = \beta \log x + \varepsilon$	Level-log regression	$\hat{\beta} = \Delta y \mid_{\frac{\Delta x}{x}=1}$	If x increases by 100 percent, then y changes by $\hat{\beta}$ units.	$x = preparation$ $y = exam\_point$
$\log y = \beta \log x + \varepsilon$	Log-log regression	$\hat{\beta} = \frac{\Delta y}{y} \mid_{\frac{\Delta x}{x}=1}$	If x increases by 1 percent, then y changes by $\hat{\beta}$ percent.	$x = wages$ $y = hours\_worked$

Wann ist es sinnvoll, die abhängige/unabhängige Variable zu transformieren? Z. B. Dann, wenn die Linearität der Parameter nicht gegeben ist (Wenn Gauss-Markov Annahmen nicht erfüllt sind)

1. Wenn Gauss-Markov-Annahme der Linearität in den Parametern verletzt sein sollte (d.h. es existiert ein linearer Zusammenhang) dann transformiert man die unabhängigen Variablen (also X-Variablen), Beispiel: Level-log model
2. Andere Annahme die verletzt sein könnte: Annahme der Homoskedastizität (bzw. Verletzung der Homoskedastizität heißt Heteroskedastizität), d.h. die Varianz der Fehlerterme nicht constant
    a. [Annahme für Homoskedastizität: Var(ui) = s^2 für alle Beobachtungen. Wenn dies verletzt ist, ist die Varianz der Fehlerterme nicht mehr constant, sondern ändert sich über die Beobachtungen. D.h. man hat Heteroskedastizität, in diesem Fall muss man die abhängige Variable transformieren (weil dann die abhängige Variable nicht symmetrisch ist). Wie findet man das raus? Histogramm von der abhängigen Variable plotten und schauen ob diese symmetrisch ist oder nicht. Wenn logarithmischer Verlauf besteht (oft bei Lohn der Fall) dann würde man die transformieren und den logarithmus nehmen. Beispiel: log-level model

Level-level interpretation: Wenn X um eine Einheit steigt, dann verändert sich Y um beta_dach Einheiten.

Log-Level interpretation (abhängige Variable transformiert): Wenn X um eine Einheit steigt, dann verändert sich Y um beta_dach * 100%

Level-Log interpretation: Wenn X um 100% steigt, dann verändert sich Y um beta_dach Einheiten

Log-log interpretation (beide Variablen werden transformiert): Wenn X um 1% steigt, dann verändert sich Y um beta_dach %

Summary: Mithilfe der linearen Regression werden Schätzer (beta_dach) für die Betas gesucht. Dies wird getan indem man die Summe der quadrierten Residuen minimiert.

R^2: man will nur Variablen im Modell haben, die auch wirklich einen Einfluss auf die Y-Variable haben, um die Gefahr des overfittings zu vermeiden. Dies erlangt man über das adjustierte R^2.

Mit subset selection methods werden modelle gefunden, deren performance dann in den Validierungsdaten getestet wird

Chapter 6 – Logit

<u>Warum verwendet man bei binären abhängigen Variablen nicht die KQ-Methode sondern die logistische Regression? 2 Gründe</u>

1. Bei binären abhängigen Variablen ist die annahme der homoskedastizität immer verletzt (es liegt immer Heteroskedastizität vor), d.h. die Varianz der Fehlerterme ändert sich über die Beobachtungen hinweg
2. Aufgrund der Inkonsistenz, d.h. bei einer binären Variable hat man als abhängige Variable Wahrscheinlichkeiten. Bei KQ-Methode würde man auf Ergebnisse Kleiner als 0 und über 1 kommen, was keinen Sinn ergibt

Given q predictors $x_1, \dots, x_q$, the **logistic response function** is defined as

$$\Pr(y = 1 \mid x_1, \dots, x_q) = p = \frac{e^{(\beta_0+\beta_1 x_1+\cdots+\beta_q x_q)}}{1 + e^{(\beta_0+\beta_1 x_1+\cdots+\beta_q x_q)}} \in [0,1]$$

- Any $(\beta_0 + \beta_1 x_1 + \cdots + \beta_q x_q) \in (-\infty, +\infty)$ is mapped into the unit interval $[0,1]$.

$P(\text{Käufer} = 1 \mid \text{Einkommen}) = F(\beta_0 + \beta_1 \cdot \text{Einkommen}) \in [0,1]$

$F(-\infty) = 0$ ✓ $F(+\infty) = 1$ ✓

im Logit: $\Lambda(z) = F(z) = \frac{e^z}{1+e^z}$ wenn $z \to -\infty$: $\frac{0}{1} \to 0$; wenn $z \to +\infty$: $\frac{e^\infty}{e^\infty} = 1$

Warum Logistische Funktion? Da es sich um Wahrscheinlichkeiten handelt, muss das Ergebnis zwischen 0 und 1 liegen. Dies wird erreicht Mit der Logistischen Funktion e^z / 1+e^z. Durch diese Funktion werden die beiden Bedingungen F(-unendlich) = 0 und F(+unendlich)=1 erfüllt. Siehe Schaubild oben.

<u>Odds</u>

Odds:= Gewinnchance

Odds sind eine relative Wahrscheinlichkeit. p/1-p, also die Wahrscheinlichkeit, dass Y gleich 1 ist, geteilt durch die Gegenwahrscheinlichkeit, also die Wahrscheinlichkeit dass Y=0 ist [P(Y=1) / 1-P(Y=0)

Beispiel:

P = P(Bayern wins | opponent = Werder) = P(Bayern gewinnt)

1-P = P(Bayern verliert | opponent = Werder) = P(Bayern verliert)

- ⇨ Wenn odds(Bayern gewinnt |opponent = Werder) = 2, dann
- ⇨ P(Bayern gewinnt) = 2 * P(Bayern verliert)
- ⇨ D.h. Gewinnen ist 2 mal so wahrscheinlich wie verlieren, wenn Werder der Gegner ist.

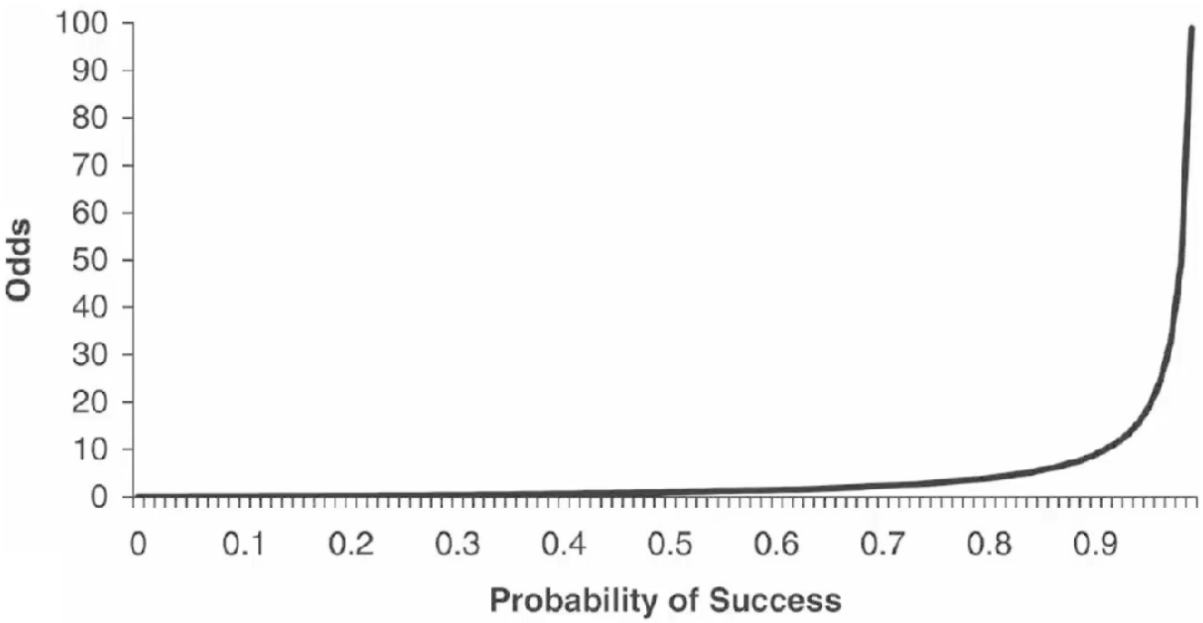
Bayern gewinnt im Schnitt 2 von 3 Spielen

odds von "Bayern gewinnt" = $\frac{2/3}{1/3} = \frac{\text{Wkt. gewinnen}}{\text{Wkt. verlieren}} = \frac{2}{1}$ 2 zu 1

Wkt. von "Bayern gewinnt" = 2/3 ≈ 66,66 %

- ➔ Absolute Wahrscheinlichkeit hingegen ist 2/3 = 66.66%

<u>Odds und Wahrscheinlichkeit zu gewinnen im Vergleich</u>



Man erkennt, dass Odds kleiner sind als die Gewinnwahrscheinlichkeit. Das liegt daran, dass man bei den Odds die Gewinnwahrscheinlichkeit noch durch die Gegenwahrscheinlichkeit teilt. Bei Gewinnwahrscheinlichkeit von 0.9 hat man z.B. Odds von 9 (weil 0.9/0.1)

Note that for

$$p = \frac{e^{(\beta_0+\beta_1 x_1+\cdots+\beta_q x_q)}}{1+e^{(\beta_0+\beta_1 x_1+\cdots+\beta_q x_q)}}$$

the odds become

$$odds = \frac{p}{1-p} = e^{(\beta_0+\beta_1 x_1+\cdots+\beta_q x_q)}.$$

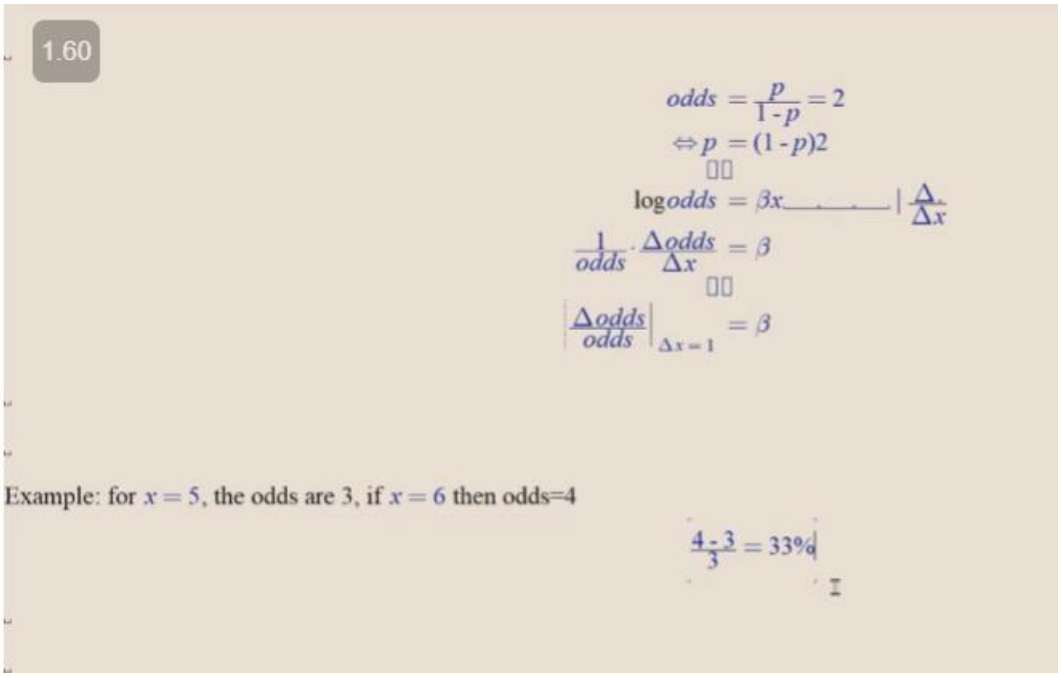
Taking logs yields the **logit**

$$\log(odds) = logit = \beta_0 + \beta_1 x_1 + \cdots + \beta_q x_q$$

So the effect of a one-unit increase in x_1 is

$$\left.\frac{\Delta odds}{odds}\right|_{\Delta x=1} = \beta_1$$

Da e^z zwar nach unten bei null begrenzt ist (Bedingung erfüllt), aber nach obenhin nicht, zieht man den log.



Es wird die change in odds angeguckt. Wenn X sich um eine Einheit erhöht (von 5 auf 6), dann ist die relative Änderung / relative change in the odds = 33% -> Dies ist der beta Wert

Simple Model: Estimation Results

Let's first start with only one predictor *income*:

$$\Pr(personal.loan = 1 \mid income) = \frac{e^{(\beta_0+\beta_1 income)}}{1+e^{(\beta_0+\beta_1 income)}}$$

$$\leftrightarrow \log[odds(personal.loan = 1 \mid income)] = \beta_0 + \beta_1 income$$

Estimation results:

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.044317   0.235591  -25.66   <2e-16 ***
income       0.036593   0.001757   20.82   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

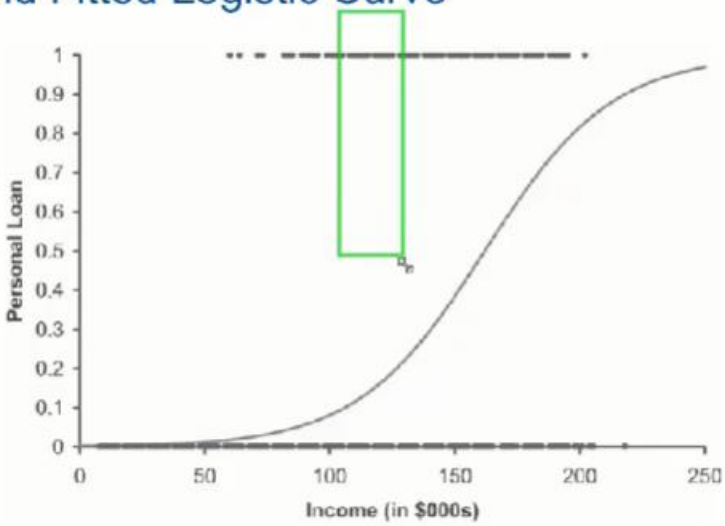
- $\hat{\beta}_1 = 0.037 = \frac{\Delta odds}{odds}$ is the predicted relative change in the odds of accepting a loan offer if the annual income increases by one unit (= 1000$).

Beta_dach = 0.037 ist die vorhergesagte relative Veränderung in den odds (also in der relativen Wahrscheinlichkeit), dass jemand einen Kredit aufnimmt, wenn das jährliche Einkommen um eine Einheit (hier = 1000$) steigt.

⇨ Wenn das Einkommen um 1000$ steigt, dann steigen die log(odds) [also der logarithmus der relativen Wkt., dass jemand einen Kredit aufnimmt], um 3.7%

Wenn Income um $1000 steigt, erhöhen sich die Odds, dass der Kredit angenommen wird, um 3.7%. ^Beta is the relative change in the odds

Data Points and Fitted Logistic Curve



The fitted logistic curve is $\widehat{\Pr}(personal.loan = 1 \mid income) = \frac{e^{(-6.044+0.037 income)}}{1+e^{(-6.044+0.037 income)}}$

➔ Die Kurve ist die vorhergesagte Wahrscheinlichkeit

Confusion Matrix

Accuracy:= gibt an wie viele 1en richtig als 1en klassifiziert wurden und wie viele 0en richtig als 0en.

Specifity:= Percentage of correctly classified 0's (negatives)

Sensitivity:= Percentage of correctly classified 1's (positives) -> (1,1) / ((0,1)+(1,1)) (Zeile,Spalte)

Wann will man Sensitivity maximieren?

zB wenn man Mailing verschicken will an diejenigen, die einen Kredit annehmen würdn. Das Modell soll also möglichst sensitiv auf diese Fälle reagieren. (Sensitivity maximieren wenn man die 1en maximieren will)

Wenn es jedoch wichtig ist, die negativen (also 0en) richtig zu klassifizieren, würde man die spezifität maximieren.

--- Categorical Variables / Dummies: One variable is left out and this variable is used as the reference category to avoid multicollinearity

Loan Acceptance Beispiel:

Results for the Full Model

```
Call:
glm(formula = personal.loan ~ ., family = "binomial", data = train.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0380  -0.1847  -0.0627  -0.0183   3.9810

Coefficients:
                               Estimate  Std. Error z value             Pr(>|z|)
(Intercept)                -12.6805628   2.2903370  -5.537          0.0000000308 ***
age                         -0.0369346   0.0848937  -0.435                0.66351
experience                   0.0490645   0.0844410   0.581                0.56121
income                       0.0612953   0.0039762  15.416 < 0.0000000000000002 ***
family                       0.5434657   0.0994936   5.462          0.0000004705 ***
ccavg                        0.2165942   0.0801900   3.599               0.00032 ***
educationGraduate            4.2681068   0.3703378  11.525 < 0.0000000000000002 ***
educationAdvanced/Professional 4.4408154 0.3723360  11.927 < 0.0000000000000002 ***
mortgage                     0.0015499   0.0007926   1.955               0.05052 .
securities.account          -1.1457476   0.3955796  -2.896               0.00377 **
cd.account                   4.5855656   0.4777696   9.598 < 0.0000000000000002 ***
online                      -0.8568074   0.2191217  -3.919          0.0000888005 ***
creditcard                  -1.2514213   0.2944767  -4.250          0.0000214111 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Koeffizienten von Logit Regression kann man nur schwer interpretieren. Was man jedoch erkennt ist, ob es einen positiven oder negativen Effekt gab und ob dieser signifikant war bzw. ist.

Beispiel income: Wenn income um eine Einheit (also 1000$) steigt, dann steigt die Wahrscheinlichkeit, dass jemand einen Kredit akzeptiert. Ebenso für Familiengröße.

Interaktionsterm: immer wenn die Gefahr besteht, dass sich unabhängige Variablen gegenseitig beeinflussen können, sollte man Interaktionsterme ausprobieren, um diesem Effekt entgegen zu wirken. Hier z.B. income und education -> neue Variable: income*education

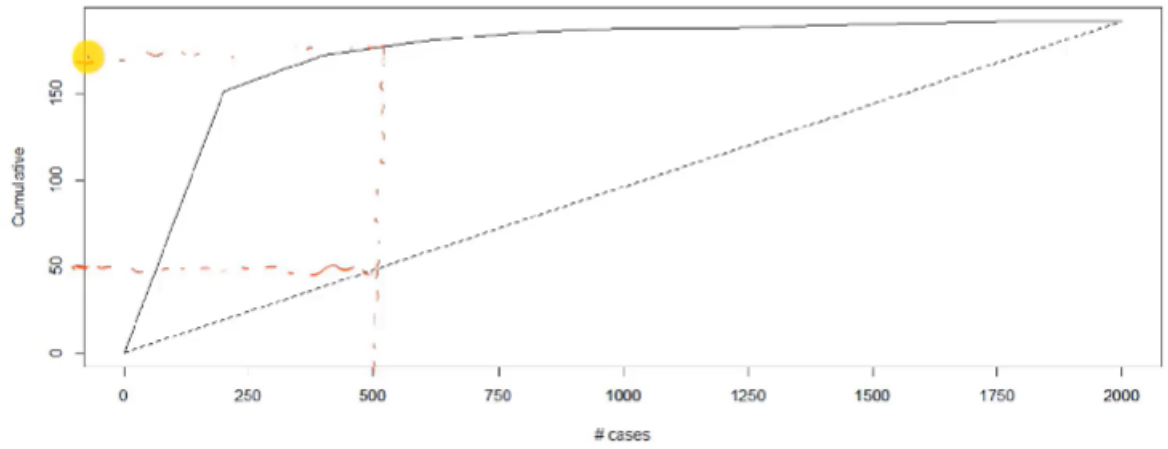
Note the statistically significant coefficients!

- If income is increasing, the odds of loan acceptance are increasing
- If family size increases by 1 unit (one additional kid), then the odds of accepting the loan offer increase by 54%
- Compared to undergrad individual (reference category, not listed here), someone with Graduate and/or advanced/professional education is way more likely to accept the loan offer (426%, 444% respectively) -> So targeting the advertisement at people with higher educational background seems worthwhile

Ways to evaluate model performance: confusion matrix and lift chart:

Lift chart

Lift chart zeigt die Verbesserung gegenüber der Naïve Rule (lift)



- ➔ Abstand zwischen Lift-curve und naive rule gibt für jede Anzahl auf der X-Achse an, wie viele zusätzliche 1en man bekommt, wenn man das Modell verwendet anstatt zu raten
- ➔ Logistische Regression sollte eher als die lineare Regression genommen werden, wenn man eine abhängige Variable hat, die 0 oder 1 sein kann. Man auch binäre Variablen mit KQ-Methode schätzen (lineares Wahrscheinlichkeitsmodell), aber dann hat man das Problem, dass man
  - 1. Heteroskedastizität hat
  - 2. Wahrscheinlichkeiten herauskriegt, die größer 1 und kleiner 0 sein können.
  - Deswegen macht Logistische Regression mehr sinn

<u>Summary:</u>

1. For every observation, the probability of accepting the loan is predicted
2. Order the data in descending order of predicted acceptance probability
3. First observation is the one with the highest acceptance probability and the last is the observation with the lowest predictive probability of acceptance

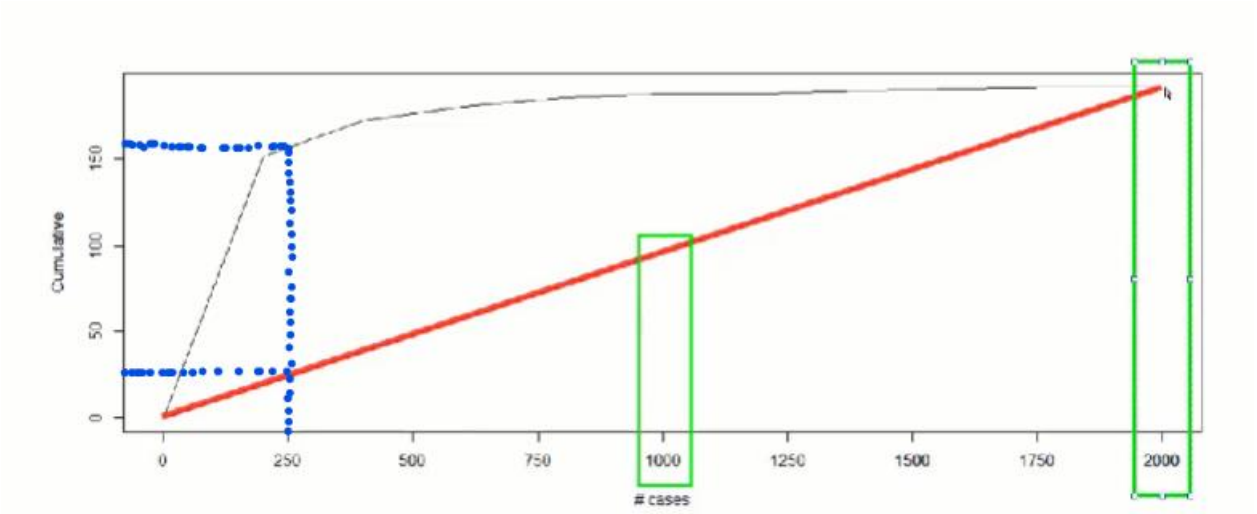
predicted probability	actual value of y	$sum(y)$	row number
99%	1	1	1
98%	1	2	2
95%	0	2	3
90%	1	3	4
...			...
2%	0	190	1999
2%	1	191	2000

4. This is then compared to the actual values (validation data) -> see the misclassifications
5. Sum(Y) = running sum of the Y -> here, 191 individuals accepted the loan offer out of 2000 (row number)
6. **Lift chart plots the running sum against the row numbers. If the probability is well predicted by the model, then (after sorting the probabilities), the running sum will be increasing with the row numbers**
7. Naïve Rule: [Number of 1's divided by N. If its 0.095 z. B., then out of 1000 randomly picked cases, one would accidently end up with 95 people with 1's/who accepted the loan

Compare the model to the Naïve Rule: When choosing a sample of N=250 for example, the model correctly predicts ~160 1's/bzw. Offer acceptings (based on the variables like age income family education etc), while randomly picking 250 out of the 2000 available observations only yields ~25 correct predictions.

-> So, picking 250 individuals identified by the model (based on descending ordered predictions) is much more efficient than just randomly picking 250 individuals from the available data (N=2000). The bank should use the model to choose which individuals

to send the loan offer to instead of randomly choosing from its customers



b) Consider the following simple logit model:

$$\log[\text{odds}(has.mortgage = 1 \mid income)] = \alpha + \beta\, income$$

i. What does β mean? Explain in detail.

ii. Do you expect $\beta > 0$, $\beta < 0$, or $\beta \approx 0$? Why?

Note: This is a theoretical question that involves no coding in Rstudio.

Odds := p / (1-p)

- ➔ If income increases, then the log odds increase by beta.
- ➔ A one-unit increase in income increases the log(odds) by beta
- ➔ This means that the odds change by 100*beta percent, i.e., the relative change in the odds is beta.

ii. How does an income increase affect the odds (p/(1-p))

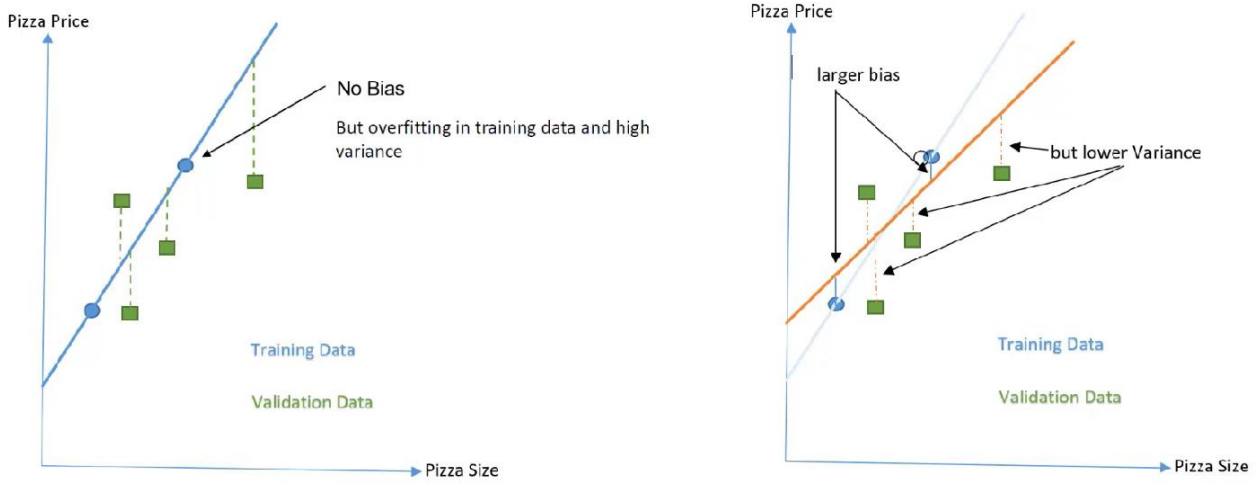
- beta > 0 is to be expected because rich people want to have multiple houses so with increasing income they are more likely to get a mortgage. Also, banks are willing to give them money (because they have a high income)
- Alternatively, beta < 0 is also possible, since the higher the income, the higher the probability that people just buy a house out of pocket without having to get a mortgage

Kapitel 7 – Lasso, Ridge and Elastic Net Regression (Shrinkage Methods)

Diese 3 Regressionsmodelle werden bei Big Data genutzt, vor allem dann, wenn Anzahl der Variablen die Anzahl der Beobachtungen übersteigt. Was sind die Grenzen von OLS oder Logit Modellen?

- ➔ Denn wenn Anzahl an Beobachtungen und Anzahl Variablen ähnlich sind (n = p) hat man bei OLS und Logit eine große Gefahr des Overfittings und dementsprechend evtl eine schlechte Vorhersagequalität
- ➔ Wenn Anzahl an Beobachtungen kleiner als Anzahl an Variablen, dann hat man nicht mehr eine eindeutige Lösung wenn man OLS/Logit benutzt, weil man hier eine unendliche Varianz hat.
- ➔ Deswegen kann man die Anzahl an eingefügten Variablen bestrafen oder lässt die irrelevanten Variablen direkt raus. Dadurch erhält man eine geringere Varianz im Validierungsdatensatz, aber dafür steigt die Verzerrung bzw. der Bias im Trainingsdatensatz, was aber nicht unbedingt schlimm ist, weil das Modell ja vor allem neue Daten (im Validierungsdatensatz) vorhersagen soll und nicht die aus dem Trainingsdatensatz
- ➔ Hierfür werden einige Koeffizienten zB auf Null gesetzt (Lasso), dadurch fliegen die Variablen aus dem Modell raus

Bias-Variance Trade-Off



- ➔ Links wurde die Regressionsgerade mithilfe von OLS gezogen. Man erkennt ein hohes Overfitting und keine Verzerrung. Aber man hat eine hohe Varianz, wenn man die Gerade mit Validation Data vergleicht.
- ➔ Rechts: Ridge Regression: Rotiert die Regressionsgerade und passt sie an die Validation Data an (hier: höheres b0 und kleineres b1 [=geringere Steigung]). Hierdurch steigt die Verzerrung etwas (Vergleiche mit Blauen Punkten von Training Data), aber dafür sinkt die Varianz (Distanz zu den grünen Punkten [=Validation Data])
    - In den Trainingsdaten sind die Koeffizienten also etwas verzerrter, aber in den Validierungsdaten ist dafür die Varianz etwas geringer -> Trade-Off. Dies ist erwünscht, da man kein Modell sucht, dass perfekt an die Trainingsdaten angepasst ist (Overfitting), sondern möglichst gut neue Daten vorhersagen kann (Validierungsdaten)
    - Man nimmt also Verzerrung in den Trainingsdaten in Kauf, um in den Validierungsdaten eine geringere Varianz zu erzielen, wodurch die Vorhersagekraft des Modells steigt

Wie funktioniert Ridge, Lasso und Elastic Net? Wie dreht man diese Gerade? Wie werden die Bestrafungsparameter eingefügt?

OLS	RIDGE	LASSO	ELASTIC NET
we min. the sum of squared residuals to find $\hat{\beta}$s	we min. the sum of squared residuals + shrinkage penalty with a **squared term** to find $\hat{\beta}$s	we min. the sum of squared residuals + shrinkage penalty with an **absolute value term** to find $\hat{\beta}$s	We combine RIDGE and LASSO
$\min_{\beta} RSS = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})$	$\min_{\beta} RSS + \lambda \sum_{j=1}^{p}\beta_j^2$	$\min_{\beta} RSS + \lambda \sum_{j=1}^{p}\lvert\beta_j\rvert$	$\min_{\beta} RSS + \lambda * \alpha * \sum_{j=1}^{p}\lvert\beta_j^L\rvert +$ $\lambda * (1-\alpha) * \sum_{j=1}^{p}\beta_j^{R^2}$

hier fehlt ei
hinter de

tuning parameter, $\lambda \geq 0$

OLS: Die Summe der quadrierten Residuen wird minimiert, um die Betas zu finden.

Ridge: Zusätzlich zu den minimierten quadrierten Residuen (min RSS von OLS), wird noch ein Bestrafungsparameter hinzugefügt:

- ➔ + Lambda mal die Summe aus den quadrierten Koeffizienten
- ➔ Tuning Parameter Lambda ist immer größer 0 und kann bis unendlich gehen
- ➔ Dieser gesamte Ausdruck (RSS+Lambda*quadrierte Koeffizienten) wird minimiert, wodurch die Betas bestimmt werden

Lasso: Ähnlich wie Ridge, nur dass statt der Summe der quadrierten Koeffizienten nun die Summe der Beträge der Koeffizienten genommen wird

Elastic Net: Ridge und Lasso werden miteinader kombiniert.

- Lasso und Ridge Bestrafung ist mit enthalten
- Alpha kann zwischen 0 und 1 sein. Wenn Alpha näher an 0 ist, gewichtet man eher die Ridge Regression, wenn Alpha näher an 1 ist, gewichtet man eher die Lasso Regression

Für Ridge, Lasso und Elastic Net, müssen die Variablen (Prädiktoren) standardisiert werden! Weil sich die Shrinkage Penalty gleich auswirkt auf alle Variablen. Deswegen müssen die Variablen standardisiert werden (damit sie auf einer Skala sind). Vor allem dann wenn man zB einen Abstand in km und Euro in Tausend drin hat, könnte es zu Fehlern kommen bei der Bestrafung

Shrinkage Method: Ridge

- ➔ Die Betas (Koeffizienten) werden so gewählt, dass sie RSS+Lamda*Summe der quadrierten Koeffizienten minimiert.
  - Shrinkage Penalty (Lambda*Summe quadrierter Koeffizienten) ist klein, wenn Betas nahe Null sind
  - Je größer Lambda (also je mehr man bestraft) desto näher sind die Koeffizienten an Null
  - Je größer Lambda, desto größer die erzielte Verzerrung, aber dafür desto kleiner die Varianz
- ➔ Ridge verwendet alle Variablen, eine Interpretation wird also noch schwieriger aufgrund der Verzerrung
- ➔ Vorteil Ridge im Vergleich zu OLS: Ridge Regression funktioniert auch wenn Anzahl der Beobachtungen kleiner ist als Variablenzahl bzw wenn n = p

Shrinkage Method: Lasso

- ➔ Bestrafung ist auch hier klein, wenn Betas nahe Null sind
- ➔ Wenn man Lambda erhöht, dreht sich die Regressionskurve: Varianz in den Validierungsdaten sinkt und Verzerrung in Trainingsdaten steigt
- ➔ Unterschied zu Ridge: Penalty zwingt einige der Koeffizienten dazu, Null zu werden, wenn Lambda genügend groß wird. Bei Ridge können die Betas nicht Null werden
- ➔ Vorteil Lasso ggü. Ridge: man kann Variablen selektieren: Modell ist einfacher zu interpretieren als bei Ridge

Bei Ridge können die Koeffizienten nicht gleich Null werden, bei Lasso aber schon. Deswegen hat man mit dem Lasso Modell tendenziell weniger Koeffizienten als beim Ridge Modell, da diese rausfallen

- ➔ **Lambda = Varianz, Beta = Verzerrung. Je größer das Lambda, desto geringer die Varianz, jedoch auch desto größer die Verzerrung. Damit das ganze minimiert werden kann, muss das Beta möglichst klein werden**

Vergleich Ridge und Lasso:

Beide nützlich wenn man mehr oder gleich viele Variablen wie Beobachtungen hat -> Big Data (wenn man extrem viele Variablen hat)

- ➔ Ridge behält alle Variablen (keine Variablenselektion wie bei Lasso, wo einige Variablen auf Null geschrumpft werden)
- ➔ Dennoch verringern beide Regressionsmodelle die Variation (Varianz)

Welche der beiden performt besser? Kommt auf die Forschungsfrage an und darauf, was für Variablen man hat. Aber Daumenregel: Man sollte Ridge benutzen, wenn die meisten Variablen aus dem Datensatz irgendwie nützlich sind und Lasso, wenn die meisten Variablen nicht so nützlich sind. -> zB bei Pizza Preis Regression: Variable zur Schuhgröße des Kochs oder Augenfarbe, etc., (wo viele nutzlose Variablen enthalten sind) würde sich Lasso anbieten. Wenn jedoch jede Variable zum Erklärungsinhalt beiträgt, sollte man auf Ridge setzen

- ⇨ Wenn man jedoch nicht weiß, wie nützlich jede Variable ist, kann man **Elastic Net Regression** benutzen

Shrinkage Method: Elastic Net

Choose β_js such, that they minimize $RSS + \lambda * \alpha * \sum_{j=1}^{p} |\beta_j^L| + \lambda * (1-\alpha) * \sum_{j=1}^{p} \beta_j^{R^2}$

- If $\lambda = 0$ => OLS
- If $\alpha = 1$ => LASSO
- If $\alpha = 0$ => RIDGE
- If $0 < \alpha < 1$ => ELASTIC NET

Combination of Ridge and Lasso

➔ Elastic Net wird verwendet wenn man nicht genau weiß wie nützlich die Variablen sind

➔ Vorteil von Elastic Net Methode: Wenn man Korrelation zwischen Variablen hat. Lasso nimmt nur eine Variable und schmeißt die anderen raus, wenn diese korreliert sind. Ridge würde alle Variablen drin lassen, aber verkleinert (schrumpft) die Koeffizienten von den korrelierten Variablen alle zusammen.

- Dies kann zu Problemen führen, wenn Korrelationen zwischen den Variablen bestehen. Elastic Net löst dies dadurch, dass von der Gruppe der korrelierten Variablen entweder alle oder gar keine der Variablen aufgenommen werden in der Regression

Welches ist das beste Shrinkage Modell?

```
> data.frame(
+   RMSE_OLS = RMSE(ols.pred, y[test]),
+   RMSE_Lasso = RMSE(lasso.pred, y[test]),
+   RMSE_Ridge = RMSE(ridge.pred, y[test]),
+   RMSE_Elastic = fit$RMSE,
+   R2_OLS = R2(ols.pred, y[test]),
+   R2_Lasso = R2(lasso.pred, y[test]),
+   R2_Ridge = R2(ridge.pred, y[test]),
+   R2_Elastic = fit$Rsquared
+ )
  RMSE_OLS RMSE_Lasso RMSE_Ridge RMSE_Elastic       R2_OLS        X1      X1.1 R2_Elastic
1  5129.76   1159.315    1165.16     1236.635 0.0004768107 0.9064943 0.9081332   0.891312
```

OLS Benchmark | Lasso | Ridge | Elastic Net

The OLS model including all potential controls performs worst

The Ridge and Lasso outperform Elastic Net and Lasso has smallest RMSE

Wie findet man das Lambda?

R probiert verschiedene Lambdas aus und checkt wie diese performen und vergleicht dann die Performance mit anderen Lambdas.

➔ Vergleich der Performance mithilfe von 10-fold cross validation:

- Trainingsdaten werden in 10 Blöcke aufgeteilt
- Aus den 10 Blöcken wird einer rausgezogen, mit den 9 anderen Blöcken werden verschiedene Lambdas getestet und mit dem herausgezogenen Block verglichen. Dann wird das Ergebnis gespeichert. Das ganze geschieht 10 Mal und aus all den Iterationen wird dann die beste Performance ausgesucht und das Lambda wird dann genommen.

Shrinkage Methods: Classification, Logit

Ridge, Lasso und Net Elastic kann auch bei Logit Regression verwendet werden:

- Schätzer im Logit Modell werden mit Maximum Likelihood Methode bestimmt. Optimale Betas werden gefunden indem man die Summe der Likelihoods maximiert. Es ergibt sich für die drei Methoden also:
  - Ridge with logistic regression: max sum of likliehoods $+\lambda \sum_{j=1}^{p} \beta_j^{R\,2}$
  - Lasso with logistic regression: max sum of likliehoods $+\lambda \sum_{j=1}^{p} |\beta_j^{L}|$
  - Elastic Net logistic regression: max sum of likliehoods $+ \alpha * \lambda \sum_{j=1}^{p} |\beta_j^{L}| + (1-\alpha) * \lambda \sum_{j=1}^{p} \beta_j^{R\,2}$
-
- Also statt die summierten quadrierten Residuen (min RSS) werden nun die likelihoods maximiert

➔ Ridge, Lasso oder Net Elastical wird nun auf die Trainingsdaten angewandt und ein optimales Modell wird gefunden

➔ Das Modell wird nun genutzt um die Vorhersage über die Wahrscheinlichkeiten im Validierungsdatensatz zu errechnen

➔ Dann werden die vorhergesagten Wahrscheinlichkeiten klassifiziert in 1er und 0er (mittels cut-off)

➔ Dann kann man Sensitivity, Specificity und Accuracy ausrechnen um zu prüfen wie gut die Klassifizierungen vom Modell getroffen wurden

Zusammenfassung Kapitel 7

Lasso, Ridge und Elastic Net Regression eignen sich bei Klassifizierungen und Vorhersagungen für Big Data

- Die drei Methoden bestrafen oder exkludieren irrelevante Variablen
- Sie erhöhen die Verzerrung im Testdatensatz, verringern dafür aber die Variation (Varianz) im Valid.datensatz
- Ridge enthält ALLE Variablen (die irrelevanten Variablen werden sehr klein bei ridge, aber nicht ganz Null -> Keine Variablenselektion)
- Lasso selektiert Variablen indem irrelevante Koeffizienten zu Null geschrumpft werden
- Elastic Net Regression kombiniert Ridge und Lasso, nützlich wenn man nicht weiß, wie viele der Variablen nützlich sind oder nicht
- Alle drei Methoden sind für Prediction und Klassifizierung gedacht: wenn man etwas über die Inferenz erfahren möchte, benötigt man Lasso für Inferenzalgorithmen

Allgemein:

Set.seed setzt den Zufallszahlengenerator auf einen bestimmten Startwert. Ermöglicht reproduzierbare Ergebnisse. Interessant zB für Debugging, oder um den Code nachvollziehbarer für andere zu machen

Warum skalieren?

Skalierung ermöglicht einen Vergleich von Daten auf gleicher Basis. Z.B. wenn man im Datensatz eine Variable in 1000€ Einheiten gemessen hat und eine andere Variable in cm, kann dies zu Schwierigkeiten beim Vergleich der Variablen führen. Die Distanzen in den Werten von den Variablen werden also vergleichbarer gemacht

Standardisieren:

Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

Die Daten werden so transformiert, dass sich eine Verteilung mit einem Mittelwert von 0 und einer Standardabweichung von 1 ergibt, sodass man die Werte aus den Daten auf der gleichen Basis vergleichen kann.

Normalisieren:

Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling.

Beim Normalisieren werden die Daten auf dieselbe Skala gebracht, sodass sie in einem Intervall von 0 bis 1 liegen.

Code für Normalisieren

```
age_norm <-rescale(age, to=c(0,1), from=range(age, na.rm=TRUE))
```

oder

```
age_norm2 <- (age-min(age))/(max(age)-min(age))
```

Normalisieren vs. Standardisieren

Die Normalisierung wird verwendet, wenn die Daten keine Gaußsche Verteilung aufweisen, während die Standardisierung bei Daten mit Gaußscher Verteilung verwendet wird.

Die Normierung skaliert in einem Bereich von [0,1]. Standardisierung ist nicht auf einen Bereich begrenzt. Bei der Standardisierung besteht die Skalierung darin, dass die Verteilung der Daten im Mittel = 0 ist und eine SD von 1 hat.

Die Normalisierung wird durch Ausreißer stark beeinträchtigt. Die Standardisierung wird durch Ausreißer leicht beeinträchtigt.

Von Normalisierung spricht man, wenn keine Annahmen über die Datenverteilung getroffen treffen. Die Standardisierung wird verwendet, wenn man Annahmen über die Datenverteilung trifft.

ROC

The ROC graph summarizes all oft he confusion matrices that each cut-off value produces

Confusion Matrix

Ist eine Tabelle die dafür genutzt wird, die Performance eines classification Modells zu evaluieren auf Basis eines Validierungsdatensatz, für den die tatsächlichen Werte bekannt sind.

- ➔ Mit sinkendem Cut-Off value steigt die sensitivity (weil mehr True Positives „eingefangen“ werden) aber dafür sinkt die specificty (weil weniger True Negatives „eingefangen“ werden)

Lift Chart

Verbesserung des Modells (geordnet nach absteigenden Wahrscheinlichkeiten auf Treffer) gegenüber einer zufälligen Ziehung aus dem Datensatz

Schätzmodell vs. Vorhersagemodell:

Schätzmodell zB: price = beta_0 + beta_1*diesel + beta_2*petrol **+ e**

Vorhersagemodell: prîce = 9421 + 1873*diesel + 1258*petrol

Vorhersagemodell ist mit Hütchen und Zahlen (wenn gegeben)
Über die Variable kommt kein Hütchen, nur über den Koeffizienten.
Im Vorhersagemodell kein + e

Woran liegt es dass Koeffizienten im erweiterten Modell steigen/sinken?

Kann daran liegen, dass die Variablen (Diesel und Benziner) in Wirklichkeit mit anderen Variablen korrelieren. Wenn man diese anderen Variablen nicht mit in das Modell aufnimmt, dann sind die irgendwo im Fehlerterm enthalten. Dann korrelieren die sich ändernden Variablen eben mit dem Fehlerterm. Das führt zu verzerrten Koeffizienten (sind entweder zu hoch oder zu niedrig).

- ➔ Beim vergleich des einfachen Modells mit erweiterten Modell ist wichtig zu schauen, welche Variablen im erweiterten Modell hinzugekommen sind. Wenn zB Diesel im einfachen Modell Koeffizient 1873.4*diesel hat und im erweiterten Modell 4293.6*diesel, könnte man Hypothese aufstellen, dass Diesel mehr km und weniger PS hat, da der negative Einfluss von hoher km und wenig PS sich „unerkannt" im einfachen Modell eingeschlichen hat bzw. dass Diesel mit diesen beiden Variablen unerkannt korreliert.
- ➔ Einfaches Modell: prîce = 9421.2 + 1873.4*diesel + 1258.1*petrol
- ➔ Erweitertes Modell: prîce = 7212.95 + 4293.6*diesel - 1782.3*petrol - 0.06*km + 88.9*hp

Exhaustive Search

search <- regsubsets(Price ~ ., data=train.df, nbest=1, nvmax=dim(train.df)[2], method="exhaustive")

Ziel: Anzahl der Variablen so reduzieren, dass nur relevante Variablen im Modell bleiben.

Linear Model Estimation

- ➔ Estimates a linear model by regressing the dependent variable wage on all remaining variables contained in job.training.df
- ➔ Lineare Regression der abhängigen Variable xx auf sämtliche unabhängige Variablen aus dem

Logit Model Estimation

- ➔ Estimates a logit model by regressing the dependent variable (binary outcome variable) on all remaining variables contained in the data set.
- ➔ Logistische Regression von der abhängigen Variable xx auf alle verbleibenden erklärenden Variablen yy

Logit Model interpretation

Can you interpret Beta1 directly as a change in the predicted probability of having a motorcycle?

- ➔ No. Beta1 gives the relative change in the log-odds.
- ➔ Nein. Das geschätzte Beta gibt nur Information über die relative Veränderung in den log(Odds), wenn xx um eine Einheit steigt

<u>Interpretation Logit Koeffizienten</u>:

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -13.939459   0.823628 -16.924  < 2e-16 ***
 age          1.701355   0.154948  10.980  < 2e-16 ***
had.contract  0.060777   0.003534  17.197  < 2e-16 ***
income        0.008079   0.008825   0.915     0.36
kids         -0.626242   0.100671  -6.221 4.95e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Mit zunehmendem Alter steigt die Wahrscheinlichkeit, eine Motorradversicherung zu haben
- With had.contract=1, the probability of having a motorcycle insurance in creases.
- Income is not significant, but if it was significant, it would mean that, with increasing income, the probability of having a motorcycle increases.
- With increasing number of kids, the probability of having a motorcycle insurance decreases-

```
Coefficients:
                                  Estimate
(Intercept)                    -12.6805628
age                             -0.0369346
experience                       0.0490645
income                           0.0612953
family                           0.5434657
ccavg                            0.2165942
educationGraduate                4.2681068
educationAdvanced/Professional   4.4408154
mortgage                         0.0015499
securities.account              -1.1457476
cd.account                       4.5855656
online                          -0.8588074
creditcard                      -1.2514213
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*'
```

Wenn das Einkommen um eine Einheit steigt die Wahrscheinlichkeit dass jemand einen Kredit akzeptiert

Wenn das Alter um ein Jahr steigt, sinkt die Wahrscheinlichkeit, dass jemand einen Kredit akzeptiert

Bei DummyVariablen (hier graduate und professional) immer im Vergleich zur Basiskategorie angeben: Wenn jemand Graduated ist, steigt die Warhscheinlichkeit einen kredit anzunehmen, im vergleich zu jemandem der keine Bildung hat

Was sind Odds?

Odds sind eine relative Wahrscheinlichkeit. p/1-p, also die Wahrscheinlichkeit, dass Y gleich 1 ist, geteilt durch die Gegenwahrscheinlichkeit, also die Wahrscheinlichkeit dass Y=0 ist

Which variable would you maximize here: accuracy, sensitivity or specificity? (asymmetrische Kosten)

- Bei Corona Test z.B. ist ein False-Negative gesellschaftlich teurer als ein False-Positive. Deswegen wird Sensitivität versucht zu maximieren, also die True Positives, sodass die False-Negative Rate minimiert wird
- Wenn man möglichst viele 0er herausfinden will, sollte man Specificity maximieren, da hierdurch die False-Positive Rate minimiert wird.

Describe() interpretieren

- Wenn Median und Mean nicht weit voneinander entfernt sind, spricht dies für Symmetrie in der Verteilung