**VL 10 Classification Problems and Logistic Regression**

**Classification Problems**

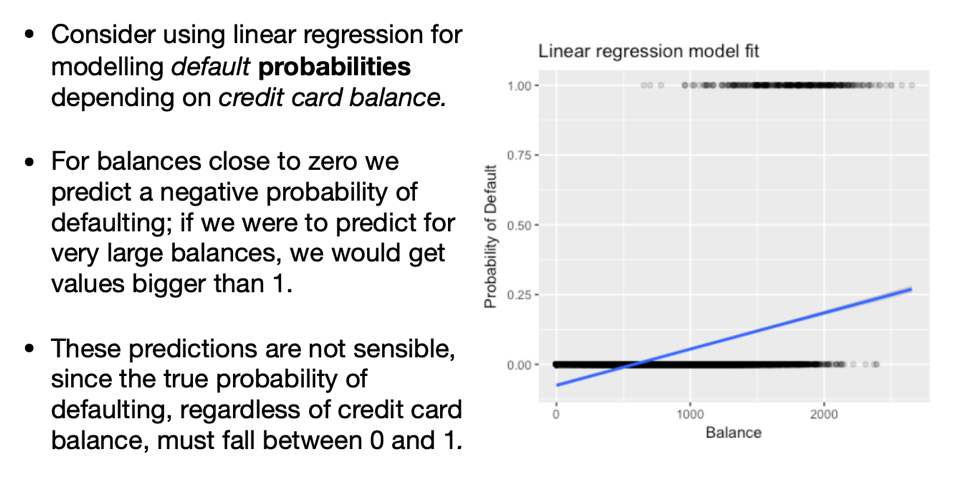
* Vorher:
* ->nur Regressionsprobleme gesehen, bei denen wir versuchen, einen kontinuierlichen Wert vorherzusagen
* Aber
* -> Die Konvention für die binäre Klassifizierung besteht darin, dass es zwei Klassen 0 und 1 gibt
* -> wie Ergebnisse wie bestanden/nicht bestanden, gewonnen/verloren, lebendig/tot oder gesund/krank darstellen
* Mit **der logistischen Regression** können wir binäre Klassifizierungsprobleme lösen
* ->bei denen wir versuchen, diskrete Kategorien vorherzusagen.
* Fälle, in denen die abhängige Variable mehr als zwei Ergebniskategorien hat, können mit multinomialer logistischer Regression analysiert werden (nicht in diesem Kurs behandelt)

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

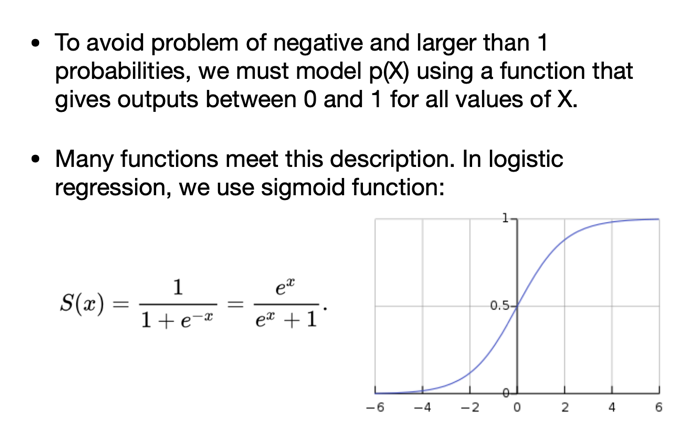
**Why not Lineare Regression?**

* Erwägung der Verwendung einer linearen Regression zur Modellierung der Ausfallwahrscheinlichkeit in Abhängigkeit vom Kreditkartensaldo
* -> Für Guthaben nahe Null sagen wir eine negative Ausfallwahrscheinlichkeit voraus
* ->würden wir eine Vorhersage für sehr hohe Guthaben treffen, würden wir Werte größer als 1 erhalten
* Diese Vorhersagen sind nicht sinnvoll
* ->da die tatsächliche Wahrscheinlichkeit eines Zahlungsausfalls unabhängig vom Kreditkartenguthaben zwischen 0 und 1 liegen muss



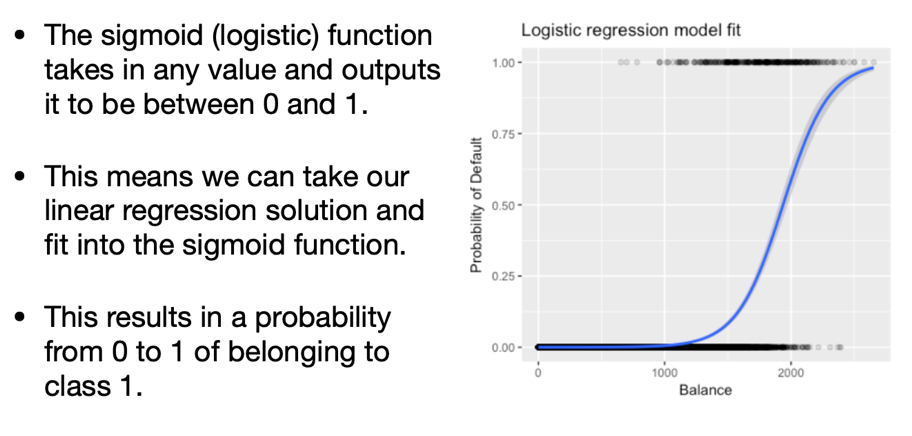
**Sigmoid Function**

* Um das Problem der negativen Wahrscheinlichkeiten und der Wahrscheinlichkeiten größer als 1 zu vermeiden
* ->muss p(X) mit einer Funktion modelliert werden, die für alle Werte von X Ausgaben zwischen 0 und 1 liefert.
* Viele Funktionen erfüllen diese Beschreibung
* ->Bei der logistischen Regression verwenden wir die Sigmoidfunktion
* *we need a function that give us sigmoid where the output is betweet 0 and 1*
* *all value up are class 1 and class 0 all which is down*
* *- eqauel means 0.5*
* *- bei Gefärhliche Sachen wie Corona thrashhold lieber runter*
* *- >dann lieber runter um alle gefährlichen Fälle*

**

**Why not Lineare Regression? Answer Sigmoid**

* Die sigmoide (logistische) Funktion nimmt einen beliebigen Wert auf und gibt ihn als Wert zwischen 0 und 1 aus.
* ->Das bedeutet, dass wir unsere lineare Regressionslösung in die Sigmoidfunktion einpassen können
* Das Ergebnis ist eine Wahrscheinlichkeit von 0 bis 1 für die Zugehörigkeit zur Klasse 1



**Model Evaluation**

* After you train a logistic regression model on some training data, you will evaluate your model’s performance on test data.
* to evaluate binary classification models you can use
* ->Confusion matrix
* -> precision/recall scores
* -> F1-Score
* -> ROC curve and more
* Confusion matrix is output of a model of one specific trashhold
* -> to evaluate for all trashhold look at ROC curve

**Confusion Matrix**

* Die Konfusionsmatrix (Konfusionstabelle) ist eine Tabelle mit zwei Zeilen und zwei Spalten
* ->in der die Anzahl der wahr-positiven, falsch-positiven, wahr-negativen und falsch-negativen Ergebnisse angegeben ist
* Bsp. Test für Zahlungsverzug bei Kreditkartenzahlungen
* -> YES - standardmäßig - positiver Test - True=1
* ->NEIN - wird nicht ausgelöst - negativer Test - Falsch=0
* - *we have two classe*
* *-> lets say yes no or no*
* *-> yes is 1 and False is 0*
* *- rows vise we have our actual output so no or yes*
* *and col vise says what our models says yes or no*

**Confusion Matrix - Basic Terminology**

* **True Positives - korrekt vorhergesagte positive Werte:**
* ->Wenn z. B. der tatsächliche Klassenwert anzeigt, dass der Kunde in Verzug geraten ist, und die vorhergesagte Klasse das Gleiche sagt
* **False Positives - falsch vorhergesagte positive Werte:**
* -> Z.B. wenn der Kunde nicht ausgefallen ist, aber die vorhergesagte Klasse sagt, dass der Kunde ausfallen wird
* **True Negatives - korrekt vorhergesagte nicht-positive Werte:**
* ->Wenn z. B. die tatsächliche Klasse sagt, dass der Kunde nicht in Verzug gerät, und die vorhergesagte Klasse dasselbe sagt
* **Falsche Negative - falsch vorhergesagte nicht-positive Werte:**
* ->Wenn z. B. der tatsächliche Klassenwert anzeigt, dass der Kunde in Verzug geraten ist, die vorhergesagte Klasse aber besagt, dass der Kunde nicht in Verzug geraten wird.
* *- TP is the right value and it has been predicted write*
* *- FP Model says it is a yes but Model says no*
* *-> it is nr 10*
* *TN it predic a No but it is a negative class so everythign right*
* *-> table nr is 50*
* *FN it belogn to 5*
* *- import for known also exam this slides*
* ***-> Wichtig für Klausur***

***Ein Bild, das Tisch enthält.

Automatisch generierte BeschreibungEin Bild, das Text, drinnen, Papier, Screenshot enthält.

Automatisch generierte Beschreibung***

****

**Confusion Matrix- Accuracy and Error Rate**

* - ACC summ up TP and TN and devide with total nr of data samples
* - error rate all wrong predicted devided all observations
* - how precision and recal are calculated
* -> Precsion

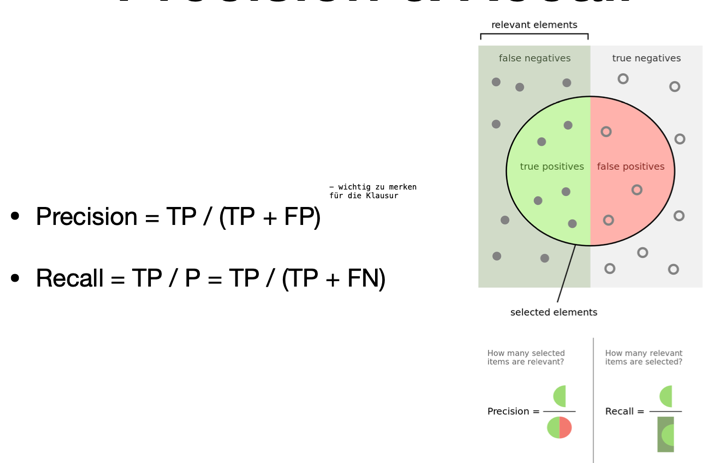
**Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung**

**Precision and Recall**

* **Precision** (positiver Vorhersagewert)
* ->Verhältnis der korrekt vorhergesagten positiven Beobachtungen zu den gesamten positiven Beobachtungen
* **->**Die Frage, die diese Metrik beantwortet, lautet: Wie viele der als säumig eingestuften Kunden sind tatsächlich säumig?
* ->Die hohe Präzision bezieht sich auf die niedrige Falsch-Positiv-Rate
* Recall (True positive Rate or Sensitivity)
* ->Verhältnis der korrekt vorhergesagten positiven Beobachtung zur gesamten tatsächlichen positiven Beobachtung
* -> Die Frage, die diese Metrik beantwortet, lautet: Wie viele der Kunden, die in Verzug geraten sind, haben wir richtig vorhergesagt?
* ->Die hohe recall quote bezieht sich auf die niedrige Falsch-Negativ-Rate

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**F1-Score**

* Der F1-Score unterscheidet nicht zwischen Precision und Recall, da Recall und Precision gleich gewichtet sind.
* -> je höher desto besser

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**General F- Score**

* Mit beta kann man sagen was größer gewichtet werde soll
* -> precision oder recall
* -> um recall wichtiger zu machen beta gleich 2 und bei precision ist es beta =0.5

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

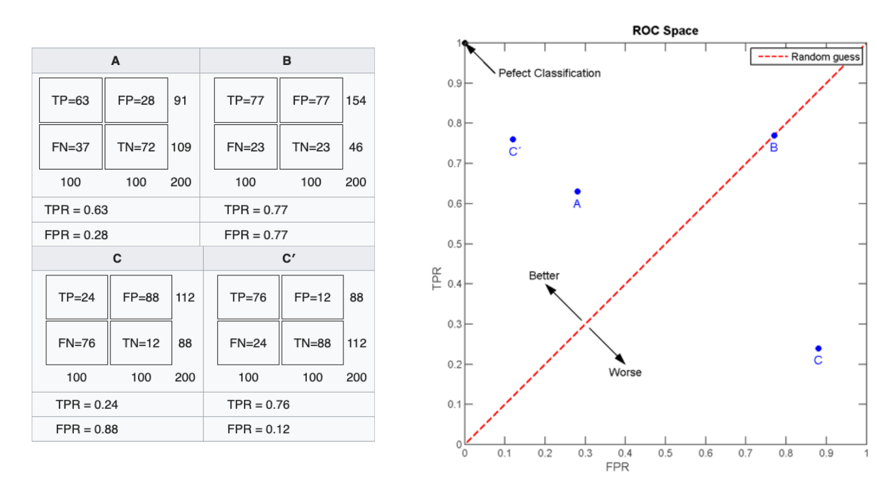
**Receiver Operating Characteristics Curve**

* Die ROC-Kurve ist eine Leistungsmessung für ein Klassifizierungsproblem bei verschiedenen Schwellenwerteinstellungen
* -> Sie gibt an, inwieweit das Modell in der Lage ist, zwischen den Klassen zu unterscheiden.
* Auf der y-Achse wird die Rate der echten Positiven (TPR oder Recall oder Sensitivität)
* ->gegen die Rate der falschen Positiven (FPR oder Falschalarmrate) auf der x-Achse aufgetragen, wobei
* **-> TPR= TP/P = TP/(TP+FN)**
* **-> FPR = FP/N. = FP/ (FP+TN)**
* The higher is the Area Under the Curve (AUC), the better is the model at distinguishing between the classes
* Die diagonal gestrichelte blaue Linie ist eine Annäherung an die Klassifizierungslinie, die auf einer zufälligen Schätzung beruht.
* - *immer auf den Context schauen und nicht AUC schauen*
* *- diagonal line says that model is guesing randomly*
* *switch outcome wenn kurve unter der Linie is*
* *je höher der AUC ist desto besser das Model*

Ein Bild, das Text enthält.

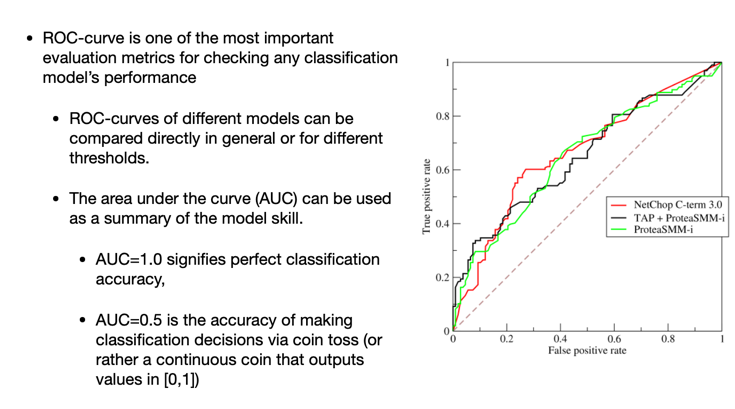
Automatisch generierte Beschreibung

**ROC- Space**

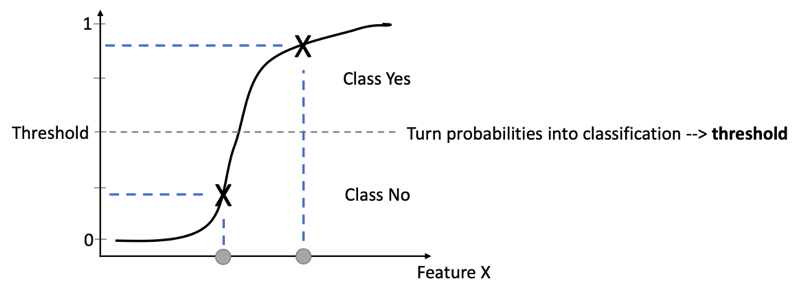
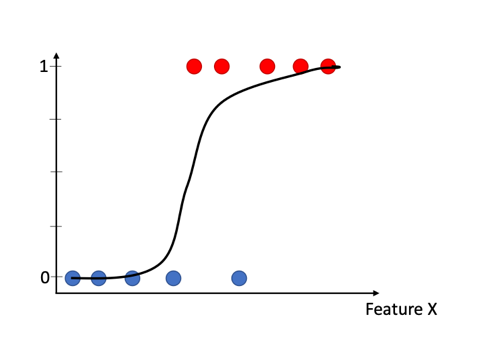


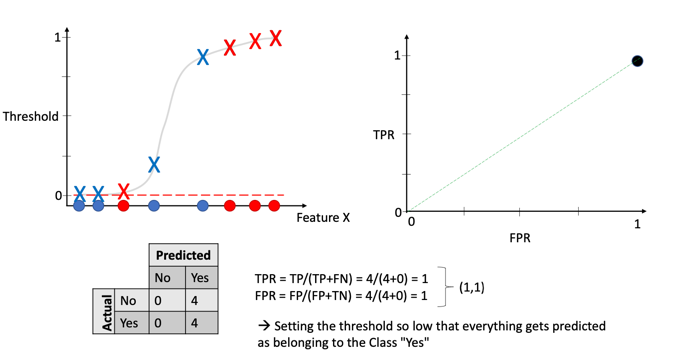
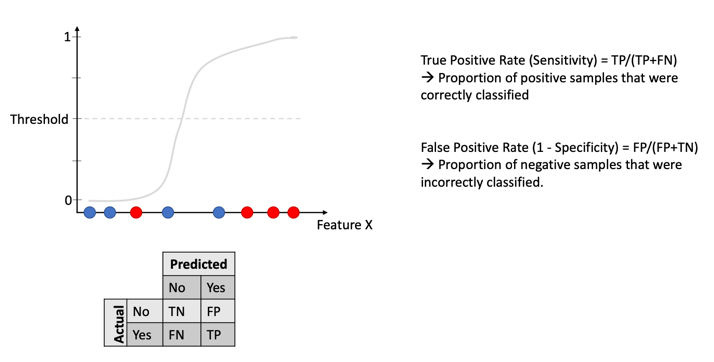
* mehre trasholds erzeugen diese Kurve

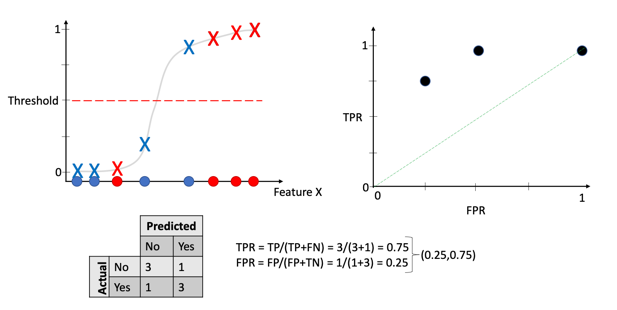
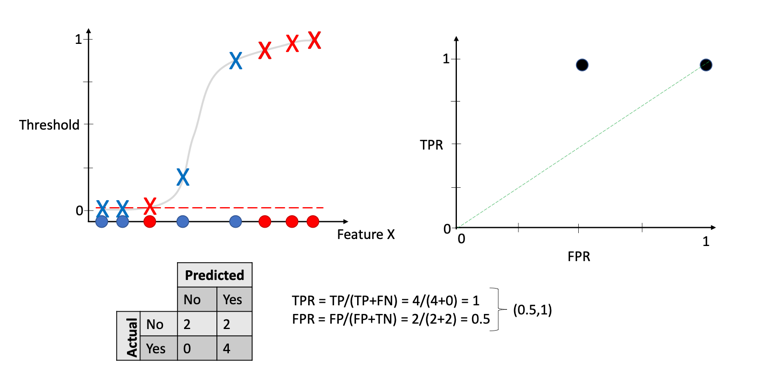
**ROC-Curve and AUC**

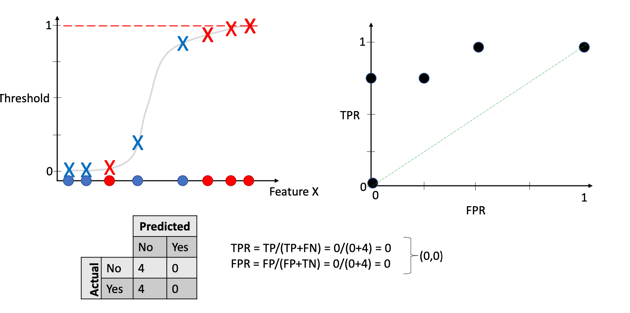
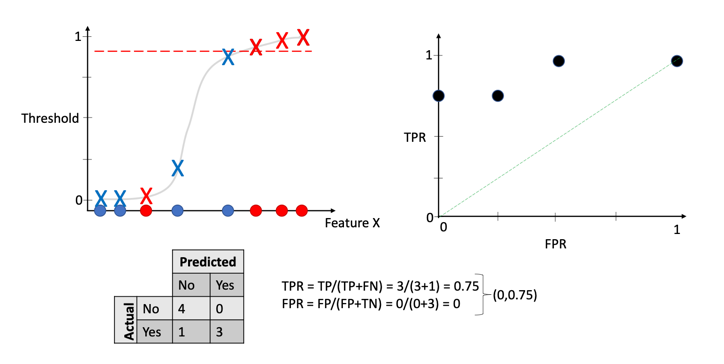


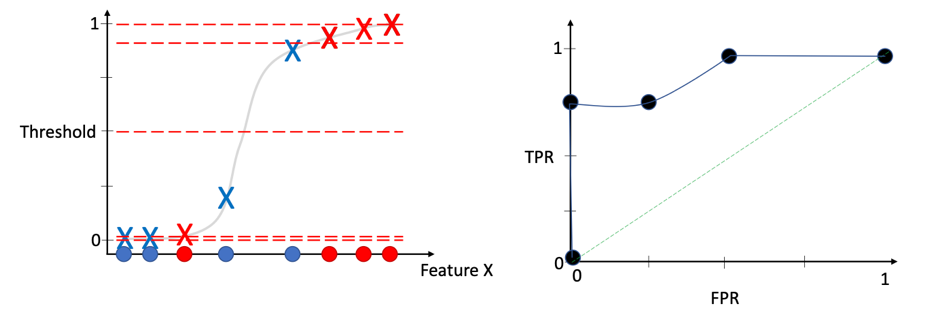
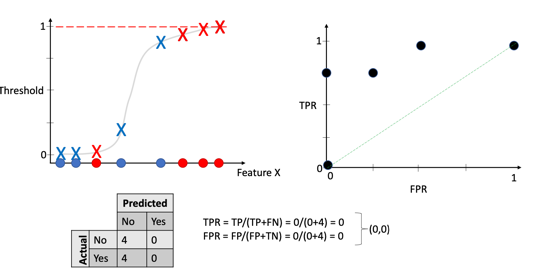
**ROC – BSP.**

****









**Logistic regression in Scikit-Learn**

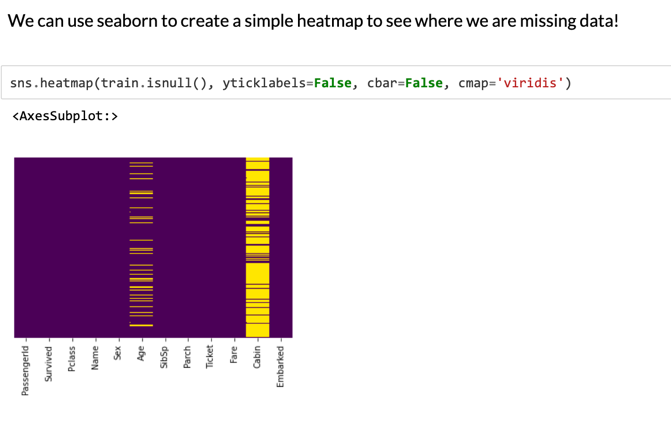
**The dataset**

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

**Check for missing data**

* #with this we can see which col has missing values

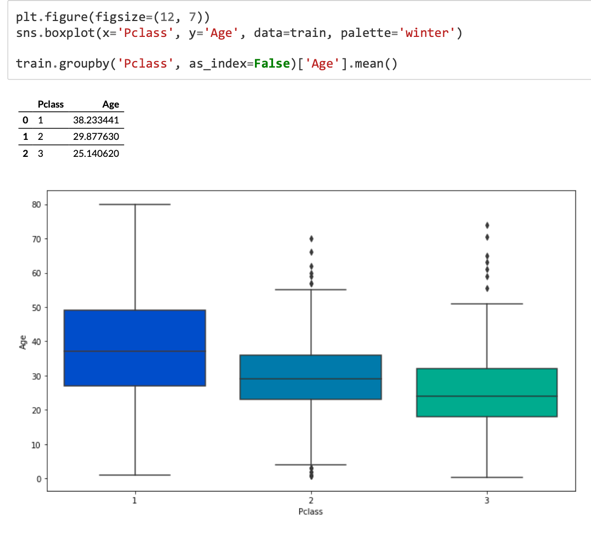
****

**Dealing with missing data**

* Ungefähr 20 % der Altersdaten fehlen.
* Der Anteil der fehlenden Altersangaben ist wahrscheinlich klein genug, um sie durch eine Form der Imputation sinnvoll zu ersetzen.
* ->Betrachtet man die Spalte "Kabine", so sieht es so aus, als ob zu viele Daten fehlen, um damit auf einer grundlegenden Ebene etwas Sinnvolles anzufangen
* ->Wahrscheinlich werden wir diese Spalte später weglassen oder sie in ein anderes Merkmal wie "Kabine bekannt: 1 oder 0" ändern.  
  (Imputation ist der Prozess des Ersetzens fehlender Daten durch Ersatzwerte)
* #there are so many missing value in cabine
* #some algo are ignoring Missing values

**Filling in missing values**

* Wir möchten fehlende Altersdaten einfügen, anstatt die fehlenden Altersdatenzeilen einfach zu streichen
* ->Eine Möglichkeit, dies zu tun, besteht darin, das Durchschnittsalter aller Passagiere einzulesen (Imputation)
* ->Wir können aber auch intelligenter vorgehen und das Durchschnittsalter nach Passagierklassen überprüfen



* # lower class means younger people
* #older people means higher class
* Wir können sehen, dass die wohlhabenderen Passagiere in den höheren Klassen tendenziell älter sind, was Sinn macht. Wir verwenden diese durchschnittlichen Alterswerte zur Imputation auf der Grundlage von Pclass for Age
* #we took the mean age for each cabine class

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

* And we see no NA in age anymore



* Later on we drop the Cabin col
* ->#löschen die col cabin weil zuviele NA vorhanden sind

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

* Let's see if there are other rows with missing value.
* ->Hierfür verwenden wir das Konstrukt **train.isna().any(axis=1),** das einen booleschen Index zurückgibt,
* ->der True ist, wenn eine entsprechende Zeile mindestens einen fehlenden Wert enthält:
* Die Methode **DataFrame.isna** enthält eine Maske von boolschen Werten für jedes Element im DataFrame, die anzeigt, ob ein Element kein NA-Wert ist.
* -Die Methode **DataFrame.any** gibt zurück, ob ein Element True ist, möglicherweise über eine Achse

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

**Converting Categorical Features**

* We'll need to convert categorical features to dummy variables using pandas!
* ->Otherwise our machine learning algorithm won't be able to directly take in those features as inputs
* For this we will rely on pandas get\_dummies() function that converts categorical variable into dummy/indicator variables.
* #convert categor into numerical
* #drop first=True sagt das man nur male sieht oder nicht
* # mit false wäre es das man male und female col sieht
* # 0 male, Female 1 oder 1 male und Female 0

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung

* Now we drop the categorical features and replace them with created 'dummy-coded' data.

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

* Now we start building our Logistic Regression Model
* -> that means we to start training our model for predicting whether a passenger survives the Titanic's accident or not.

**Train Test Split**

* First we start by splitting our data into a training set and test set.
* #surved wollen wir predcited
* #train.drop are independet X value
* #tran surv
* **-> Klausur wichtig**

Ein Bild, das Text enthält.

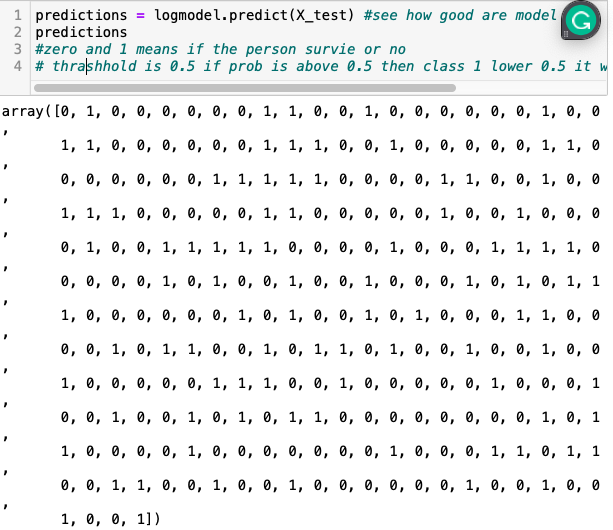
Automatisch generierte Beschreibung

**Train and Predicting**

**Ein Bild, das Text enthält.

Automatisch generierte Beschreibung**

* We are training with fit and create before a empty model
* Then we collect predictions for the testing se in predictions series
* -> we use predic on X\_test
* #see how good our model is
* #zero and 1 means if the person survie or no
* # thrashhold is 0.5 if prob is above 0.5 then class 1 lower 0.5 it will be classified to 0



**Evaluation**

* We can compute precision, recall, f1-score using a single call to classification\_report function!
* # wir sagen bei 0 schauen wir klasse 0
* #134 sind TP und Precision wird berechnet 134 /134 +42
* # Recal : 134 /134+20
* #bei klasse 1 hat man TP berechnen

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

* from sklearn.metrics import classification\_report, confusion\_matrix, ConfusionMatrixDisplay
* confusion\_matrix(y\_test,predictions)



**ROC-Curve**

* Verwenden wir zunächst die Methode predict\_proba unseres LogisticRegression-Modells, um Wahrscheinlichkeitsschätzungen für die Klasse zu erhalten, die wir vorhersagen wollen: überlebt, die mit 1 gekennzeichnet ist.
* #we need prob of one single instance if it belongs to class 0 and 1
* #also die wahrscheinlichkeit ob es zu kalsse 0 oder Klasse 1 gehör

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

* Alsp let's compute False Positive Rate (FPR), True Positive Rate (TPR) and Area Under the Curve using roc\_curve and auc functions from sklearn.metrics, where

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

* #roc curve hat 3 parameter
* #es bekommt die test labels und die vorhergesagen wahrscheinlichkeiten
* #input wäre fpr, tpr und trhash
* Bei trashhold
* -># erster wert ist das größte werte und addiert es mit 1
* #(kann man ignorieren nicht so wichtig)

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

* Am ende
* ->Plot the results using matplotlib and display AUC value in the legend

