DATA SCIENCE
ABSCHLUSSPROJEKT

HUMAN RESSOURCE
-
DATENSATZ IBM

VORGEHENSWEISE:

- ✔ Datensatz beschaffen → kostenloser Kaggle Datensatz IBM
- ✔ Deskriptive Analyse des Datensatzes
- ✔ Daten bereinigen
- ✔ Korrelationen einsehen und visualisieren
- ✔ Visualisierungen in Tableau
- ✔ Visualisierung des Datensatzes - Plots
- ✔ Supervised Learning
- ✔ Unsupervised Learning

DATENSATZ:

VORÜBERLEGUNGEN:

- Bei der Auswahl des Datensatzes habe ich mich für den Bereich „Human Ressource - Employment“ entschieden

- Employment: Unternehmen haben sich in Ihrer Struktur in den vergangenen Jahren stark verändert. Arbeitnehmer, aber auch Arbeitgeber handeln beim Thema Kündigungen kurzfristiger als in der Vergangenheit. Die Grunde dafür sind vielfältig. Folglich wird es in Zukunft für ein Unternehmen wichtiger sein, besser beurteilen zu können, wann ein Mitarbeiter aller Wahrscheinlichkeit nach kündigen wird, um möglichst effizient zu bleiben

ZIEL:

➔ Ziel dieses Projektes ist es, anhand verschiedener Algorithmen aus dem Machine-Learning-Bereich vorherzusagen, ob ein Mitarbeiter aufgrund seiner individuellen Gegebenheiten im Unternehmen in naher Zukunft kündigen würde oder nicht

Der Datensatz enthält 33 features und hat 1470 Zeilen und liefert uns mit dem Label „Attrition“ Informationen, ob ein Mitarbeiter kündigt oder nicht:

FEATURES:

Age	das Alter des Mitarbeiters
Attrition	gibt an, ob der Mitarbeiter gekündigt hat oder nicht
BusinessTravel	ob der Mitarbeiter geschäftlich unterwegs war oder nicht
Department	in welcher Abteilung der Mitarbeiter beschäftigt war
DistanceFromHome	die Entfernung von zu Hause, um den Arbeitsplatz zu erreichen
Gender	Geschlecht des Mitarbeiters
JobInvolvement	die Beteiligungsbewertung eines Mitarbeiters an der bearbeiteten Aufgabe
JobLevel	Ebene, auf der der Mitarbeiter arbeitet
JobRole	die Rolle und Verantwortlichkeit des Mitarbeiters
JobSatisfaction	Zufriedenheits-Bewertung des Mitarbeiters mit der Stelle
MaritalStatus	Familienstand des Mitarbeiters
MonthlyIncome	Monatliches Einkommen des Mitarbeiters
NumCompaniesWorked	Anzahl der Unternehmen, für die der Mitarbeiter gearbeitet hat
OverTime	ob Überstunden gemacht werden oder nicht
PercentSalaryHike	prozentuale Gehaltserhöhung seit ihrer Einstellung im Unternehmen
PerformanceRating	Leistungsbewertung

StockOptionLevel	Aktienoption
TotalWorkingYears	Gesamtarbeitsjahre des Mitarbeiters
TrainingTimesLastYear	wie viele Fortbildungen hat der Mitarbeiter absolviert
YearsAtCompany	gearbeitete Jahre im derzeitigen Unternehmen
YearsSinceLastPromotion	Zeit in Jahren seit der letzten Beförderung
YearsWithCurrManager	Jahre, in denen der Mitarbeiter unter dem aktuellen Manager arbeitet
Higher_Education	Bildungsgrad
Date_of_Hire	Einstellungsdatum des Mitarbeiters
Date_of_termination	Datum der Kündigung
Status_of_leaving	Grund für die Kündigung
Mode_of_work	Homeoffice oder Büro
Leaves	Gesamtzahl der zulässigen Urlaubstage des Mitarbeiters
Absenteeism	Gesamtzahl der Abwesenheitstage des Mitarbeiters
Work_accident	Arbeitsunfall, falls vorhanden
Source_of_hire	Anwerbungsart
Job_Mode	Vollzeit-/Teilzeit- oder Vertragsarbeit

DESKRIPTIVE ANALYSE DES DATENSATZES:

- ✗ Ausschnitt aus dem Datensatz:

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Gender	JobInvolvement	JobLevel	JobRole	JobSatisfaction	...	Date_of_Hire	Date_of
0	37	Yes	Travel_Rarely	Research & Development	2	Male	2	1	Laboratory Technician	3	...	21-01-2021	
1	21	No	Travel_Rarely	Research & Development	15	Male	3	1	Research Scientist	4	...	13-03-2021	
2	45	No	Travel_Rarely	Research & Development	6	Male	3	3	Research Director	1	...	23-01-2021	
3	23	No	Travel_Rarely	Sales	2	Male	3	1	Sales Representative	1	...	25-04-2021	
4	22	No	Travel_Rarely	Research & Development	15	Female	3	1	Laboratory Technician	4	...	14-06-2021	
5	19	Yes	Travel_Rarely	Sales	22	Male	3	1	Sales Representative	3	...	14-04-2021	
6	19	Yes	Travel_Frequently	Sales	1	Female	1	1	Sales Representative	1	...	12-01-2021	

✗ Der Datensatz hat Nullwerte in der Spalte „Unnamed: 32“, die keine Daten enthalten und außerdem 3 weitere für die Vorhersage meiner Meinung nach irrelevante Spalten „Date_of_hire“,„Date_of_termination“ und „StockOptionLevel“, die gelöscht werden können:

```
1 mitarbeiter02.drop(mitarbeiter02[["Date_of_termination", "Unnamed: 32"]], axis = 1, inplace=True)
```

```
1 mitarbeiter02.drop(mitarbeiter02[["Date_of_Hire", "StockOptionLevel"]], axis = 1, inplace=True)
```

```
1 mitarbeiter02.shape
```

```
(1470, 29)
```

✗ somit verbleiben 29 features

✗ der Datensatz hat einige kategorische Werte, die in numerische umgewandelt werden:

KATEGORISCHE WERTE - OBJECTS

0	Age	1470 non-null	int64
1	Attrition	1470 non-null	object
2	BusinessTravel	1470 non-null	object
3	Department	1470 non-null	object
4	DistanceFromHome	1470 non-null	int64
5	Gender	1470 non-null	object
6	JobInvolvement	1470 non-null	int64
7	JobLevel	1470 non-null	int64
8	JobRole	1470 non-null	object
9	JobSatisfaction	1470 non-null	int64
10	MaritalStatus	1470 non-null	object
11	MonthlyIncome	1470 non-null	int64
12	NumCompaniesWorked	1470 non-null	int64
13	OverTime	1470 non-null	object
14	PercentSalaryHike	1470 non-null	int64
15	PerformanceRating	1470 non-null	int64
16	StockOptionLevel	1470 non-null	int64
17	TotalWorkingYears	1470 non-null	int64
18	TrainingTimesLastYear	1470 non-null	int64
19	YearsAtCompany	1470 non-null	int64
20	YearsSinceLastPromotion	1470 non-null	int64
21	YearsWithCurrManager	1470 non-null	int64
22	Higher_Education	1470 non-null	object
23	Date_of_Hire	1470 non-null	object
24	Date_of_termination	0 non-null	float64
25	Status_of_leaving	1470 non-null	object
26	Mode_of_work	1470 non-null	object
27	Leaves	1470 non-null	int64
28	Absenteeism	1470 non-null	int64
29	Work_accident	1470 non-null	object
30	Source_of_Hire	1470 non-null	object
31	Job_mode	1470 non-null	object
32	Unnamed: 32	0 non-null	float64

NULLWERTE:

Age	0
Attrition	0
BusinessTravel	0
Department	0
DistanceFromHome	0
Gender	0
JobInvolvement	0
JobLevel	0
JobRole	0
JobSatisfaction	0
MaritalStatus	0
MonthlyIncome	0
NumCompaniesWorked	0
OverTime	0
PercentSalaryHike	0
PerformanceRating	0
StockOptionLevel	0
TotalWorkingYears	0
TrainingTimesLastYear	0
YearsAtCompany	0
YearsSinceLastPromotion	0
YearsWithCurrManager	0
Higher_Education	0
Date_of_Hire	0
Date_of_termination	1470
Status_of_leaving	0
Mode_of_work	0
Leaves	0
Absenteeism	0
Work_accident	0
Source_of_Hire	0
Job_mode	0
Unnamed: 32	1470

DATEN BEREINIGEN:

✗ Umwandeln der kategorischen features in numerische Werte:

```
# Gender
mitarbeiter02.Gender.replace({"Female" : 0, "Male" : 1}, inplace = True)
```

```
# Kündigung
mitarbeiter02.Attrition.replace({"No" : 0, "Yes" : 1}, inplace = True)
```

```
# Geschäftsreisen
mitarbeiter02.BusinessTravel.replace({"Non-Travel" : 0, "Travel_Rarely" : 1, "Travel_Frequently" : 2}, inplace = True)
```

```
# Department
mitarbeiter02.Department.replace({"Human Resources" : 0, "Sales" : 1, "Research & Development" : 2}, inplace = True)
```

```
# JobRole
mitarbeiter02.JobRole.replace({"Sales Executive" : 0, "Research Scientist" : 1, "Laboratory Technician" : 2,
                               "Manufacturing Director" : 3, "Healthcare Representative" : 4, "Manager" : 5,
                               "Sales Representative" : 6, "Research Director" : 7, "Human Resources" : 8,
                               }, inplace = True)
```

```
# MaritalStatus
mitarbeiter02.MaritalStatus.replace({"Single" : 0, "Divorced" : 1, "Married" : 2}, inplace = True)
```

```python
# Overtime - Überstunden
mitarbeiter02.OverTime.replace({"No" : 0, "Yes" : 1}, inplace = True)
```

```python
# Work_accident
mitarbeiter02.Work_accident.replace({"No" : 0, "Yes" : 1}, inplace = True)
```

```python
# Ausbildung
mitarbeiter02.Higher_Education.replace({"12th" : 0, "Graduation" : 1, "Post-Graduation" : 2, "PHD" : 3},
                                       inplace = True)
```

```python
# # Kündigungsgrund
mitarbeiter02.Status_of_leaving.replace({"Dept.Head" : 0, "Salary" : 1, "Work Environment" : 2, "Work Accident" : 3,
                                         "Better Opportunity" : 4}, inplace = True)
```

```python
# Arbeitsort
mitarbeiter02.Mode_of_work.replace({"WFH" : 0, "OFFICE" : 1}, inplace = True)
```

```python
# Anwerbungsart
mitarbeiter02.Source_of_Hire.replace({"Recruiter" : 0, "Job Event" : 1, "Walk-in" : 2, "Job Portal" : 3},
                                     inplace = True)
```

```python
# Jobart
mitarbeiter02.Job_mode.replace({"FullTime" : 0, "Contract" : 1, "Part Time" : 2}, inplace = True)
```

✗ bereinigter Datensatz ohne Nullwerte und mit ausschließlich numerischen Daten:

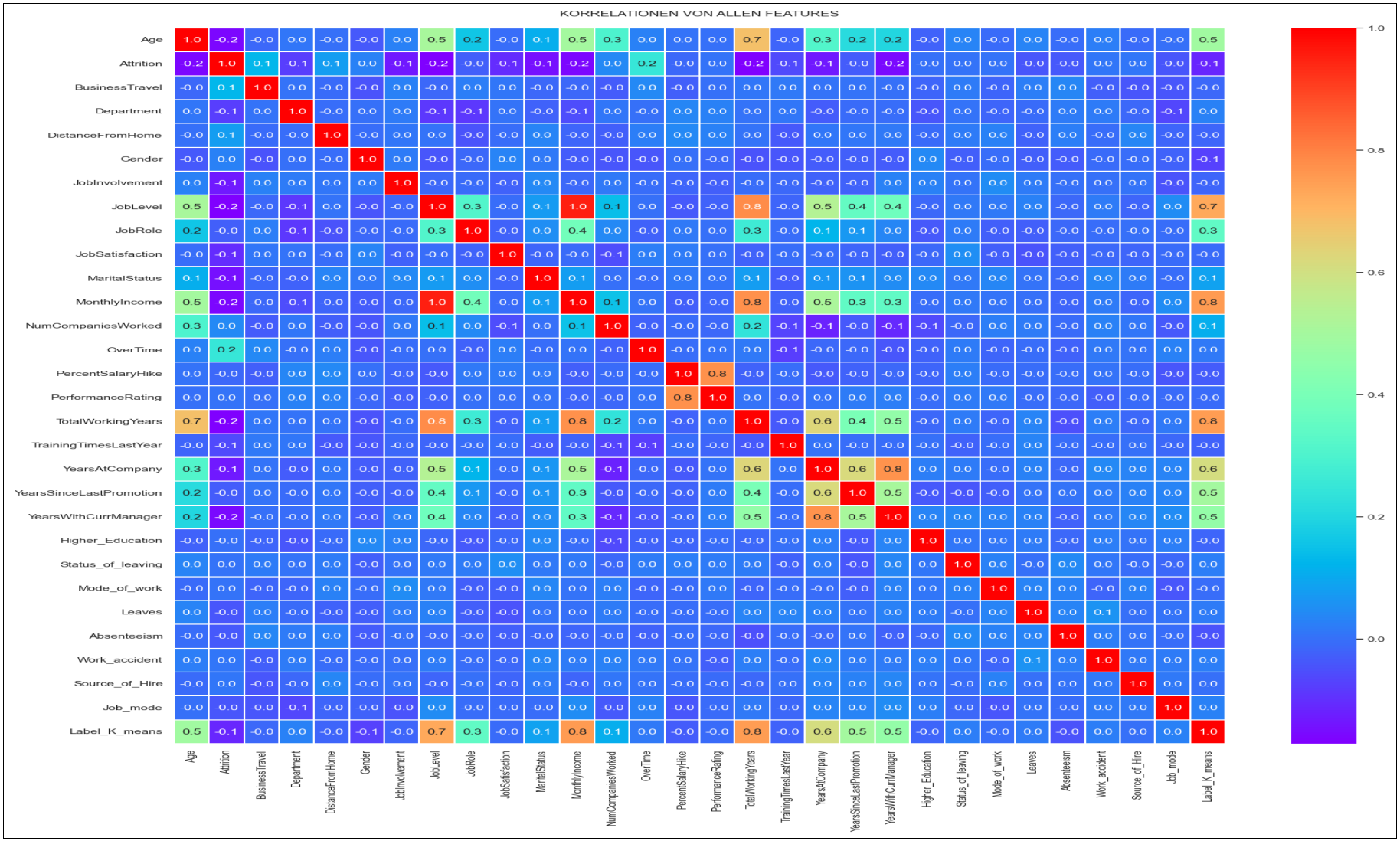
```
Data columns (total 29 columns):
 #    Column                    Non-Null Count  Dtype
---   ------                    --------------  -----
 0    Age                       1470 non-null   int64
 1    Attrition                 1470 non-null   int64
 2    BusinessTravel            1470 non-null   int64
 3    Department                1470 non-null   int64
 4    DistanceFromHome          1470 non-null   int64
 5    Gender                    1470 non-null   int64
 6    JobInvolvement            1470 non-null   int64
 7    JobLevel                  1470 non-null   int64
 8    JobRole                   1470 non-null   int64
 9    JobSatisfaction           1470 non-null   int64
 10   MaritalStatus             1470 non-null   int64
 11   MonthlyIncome             1470 non-null   int64
 12   NumCompaniesWorked        1470 non-null   int64
 13   OverTime                  1470 non-null   int64
 14   PercentSalaryHike         1470 non-null   int64
 15   PerformanceRating         1470 non-null   int64
 16   TotalWorkingYears         1470 non-null   int64
 17   TrainingTimesLastYear     1470 non-null   int64
 18   YearsAtCompany            1470 non-null   int64
 19   YearsSinceLastPromotion   1470 non-null   int64
 20   YearsWithCurrManager      1470 non-null   int64
 21   Higher_Education          1470 non-null   int64
 22   Status_of_leaving         1470 non-null   int64
 23   Mode_of_work              1470 non-null   int64
 24   Leaves                    1470 non-null   int64
 25   Absenteeism               1470 non-null   int64
 26   Work_accident             1470 non-null   int64
 27   Source_of_Hire            1470 non-null   int64
 28   Job_mode                  1470 non-null   int64
dtypes: int64(29)
```

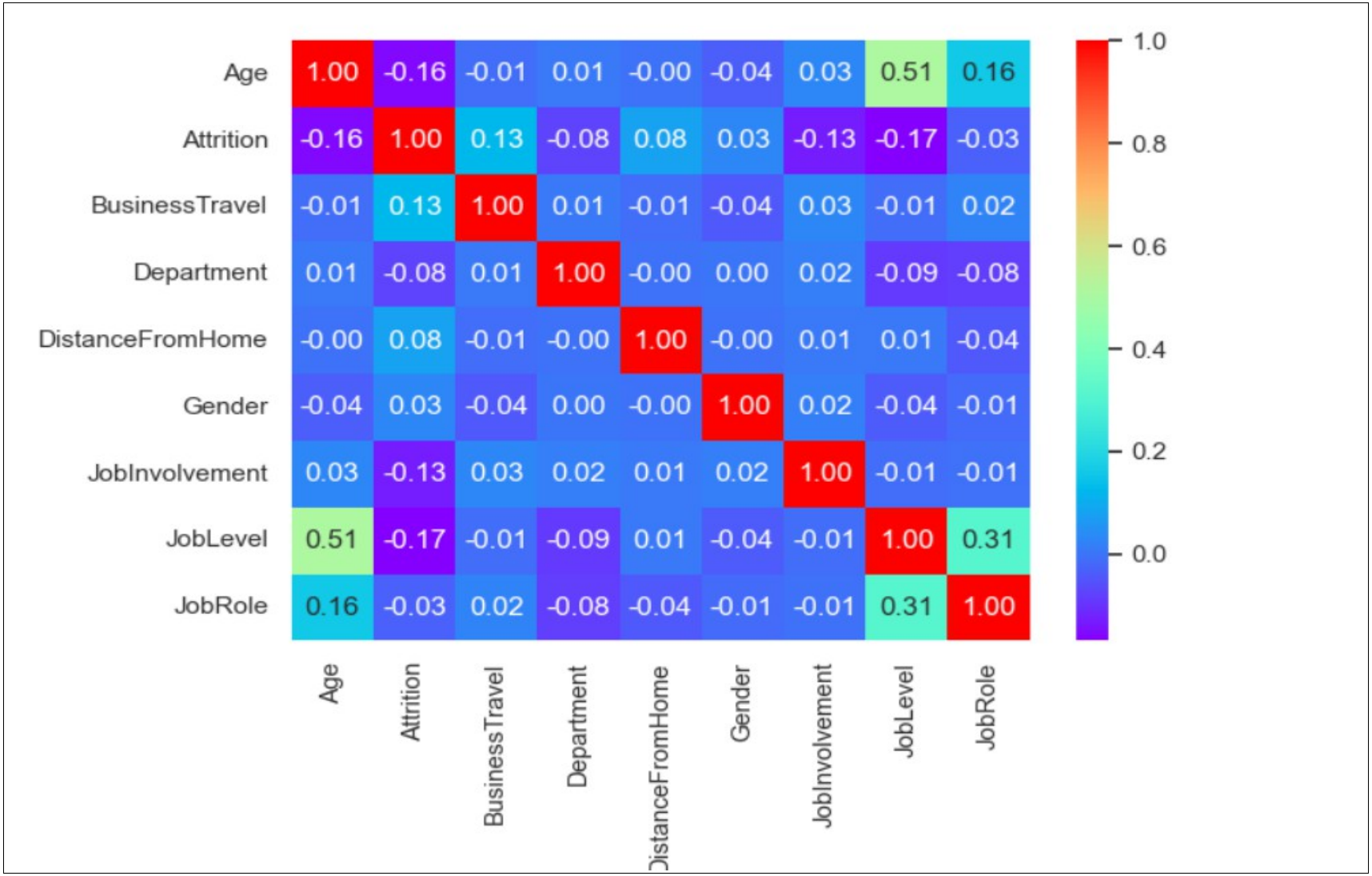
✗ deskriptive Analyse:

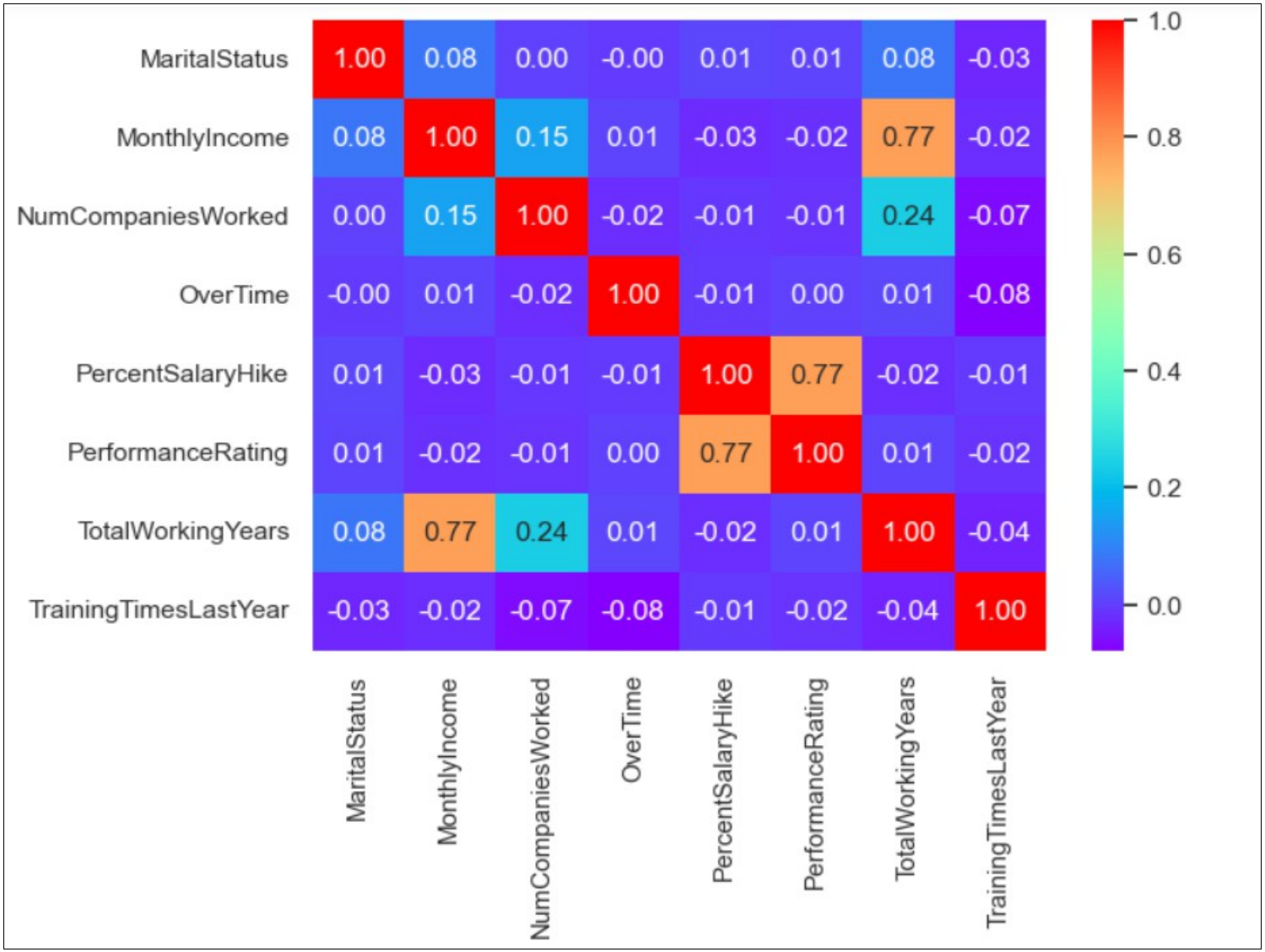
	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Gender	JobInvolvement	JobLevel	JobRole	JobSatisfaction	...	Ye
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	...	
mean	36.923810	0.161224	1.086395	1.610884	9.192517	0.600000	2.729932	2.063946	2.553061	2.728571	...	
std	9.135373	0.367863	0.532170	0.568893	8.106864	0.490065	0.711561	1.106940	2.323902	1.102846	...	
min	18.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000	1.000000	0.000000	1.000000	...	
25%	30.000000	0.000000	1.000000	1.000000	2.000000	0.000000	2.000000	1.000000	1.000000	2.000000	...	
50%	36.000000	0.000000	1.000000	2.000000	7.000000	1.000000	3.000000	2.000000	2.000000	3.000000	...	
75%	43.000000	0.000000	1.000000	2.000000	14.000000	1.000000	3.000000	3.000000	4.000000	4.000000	...	
max	60.000000	1.000000	2.000000	2.000000	29.000000	1.000000	4.000000	5.000000	8.000000	4.000000	...	

KORRELATIONEN EINSEHEN UND VISUALISIEREN:

- ✗ Korrelationen:




Age 1.00 -0.16 -0.01 0.01 -0.00 -0.04 0.03 0.51 0.16
Attrition -0.16 1.00 0.13 -0.08 0.08 0.03 -0.13 -0.17 -0.03
BusinessTravel -0.01 0.13 1.00 0.01 -0.01 -0.04 0.03 -0.01 0.02
Department 0.01 -0.08 0.01 1.00 -0.00 0.00 0.02 -0.09 -0.08
DistanceFromHome -0.00 0.08 -0.01 -0.00 1.00 -0.00 0.01 0.01 -0.04
Gender -0.04 0.03 -0.04 0.00 -0.00 1.00 0.02 -0.04 -0.01
JobInvolvement 0.03 -0.13 0.03 0.02 0.01 0.02 1.00 -0.01 -0.01
JobLevel 0.51 -0.17 -0.01 -0.09 0.01 -0.04 -0.01 1.00 0.31
JobRole 0.16 -0.03 0.02 -0.08 -0.04 -0.01 -0.01 0.31 1.00
Age
Attrition
BusinessTravel
Department
DistanceFromHome
Gender
JobInvolvement
JobLevel
JobRole
1.0
0.8
0.6
0.4
0.2
0.0


MaritalStatus 1.00 0.08 0.00 -0.00 0.01 0.01 0.08 -0.03
MonthlyIncome 0.08 1.00 0.15 0.01 -0.03 -0.02 0.77 -0.02
NumCompaniesWorked 0.00 0.15 1.00 -0.02 -0.01 -0.01 0.24 -0.07
OverTime -0.00 0.01 -0.02 1.00 -0.01 0.00 0.01 -0.08
PercentSalaryHike 0.01 -0.03 -0.01 -0.01 1.00 0.77 -0.02 -0.01
PerformanceRating 0.01 -0.02 -0.01 0.00 0.77 1.00 0.01 -0.02
TotalWorkingYears 0.08 0.77 0.24 0.01 -0.02 0.01 1.00 -0.04
TrainingTimesLastYear -0.03 -0.02 -0.07 -0.08 -0.01 -0.02 -0.04 1.00
MaritalStatus
MonthlyIncome
NumCompaniesWorked
OverTime
PercentSalaryHike
PerformanceRating
TotalWorkingYears
TrainingTimesLastYear
1.0
0.8
0.6
0.4
0.2
0.0

WICHTIGE KORRELATIONEN:

KORRELATION 1.0:

- MonthlyIncome - JobLevel

KORRELATION 0.8:

- YearsAtCompany - YearsWithCurrentManager
- PerformanceRating - PercentSalaryHike
- MonthlyIncome - TotalWorkingYears
- JobLevel - TotalWorkingYears

KORRELATION 0.7:

- Age - TotalWorkingYears

KORRELATION 0.6:

- YearsAtCompany - YearsSinceLastPromotion
- TotalWorkingYears - YearsAtCompany

HÖCHSTE KORRELATIONEN MIT DEM LABEL ATTRITION:

- OverTime - Attrition 0.25
- BusinessTravel - Attrition 0.13
- DistanceFromHome - Attriton 0.08
- JobLevel - Attrition -0.17
- TotalWorkingYears - Attrition -0,17
- Age - Attrition -0.16
- YearsWithCurrentManager - Attrition -0.16
- MaritalStatus - Attrition -0.15

✗ Korrelationen Label „Attrition“ mit allen features:

	index	Attrition
0	OverTime	0.25
1	BusinessTravel	0.13
2	DistanceFromHome	0.08
3	NumCompaniesWorked	0.04
4	Gender	0.03
5	Source_of_Hire	0.01
6	Work_accident	0.01
7	Mode_of_work	0.01
8	Status_of_leaving	0.01
9	Higher_Education	-0.00
10	PerformanceRating	0.00
11	PercentSalaryHike	-0.01
12	YearsSinceLastPromotion	-0.03
13	JobRole	-0.03
14	Leaves	-0.04

	index	Attrition
15	Absenteeism	-0.04
16	Job_mode	-0.04
17	TrainingTimesLastYear	-0.06
18	Department	-0.08
19	JobSatisfaction	-0.10
20	YearsAtCompany	-0.13
21	JobInvolvement	-0.13
22	MaritalStatus	-0.15
23	YearsWithCurrManager	-0.16
24	MonthlyIncome	-0.16
25	Age	-0.16
26	TotalWorkingYears	-0.17
27	JobLevel	-0.17

VISUALISIERUNGEN IN TABLEAU:

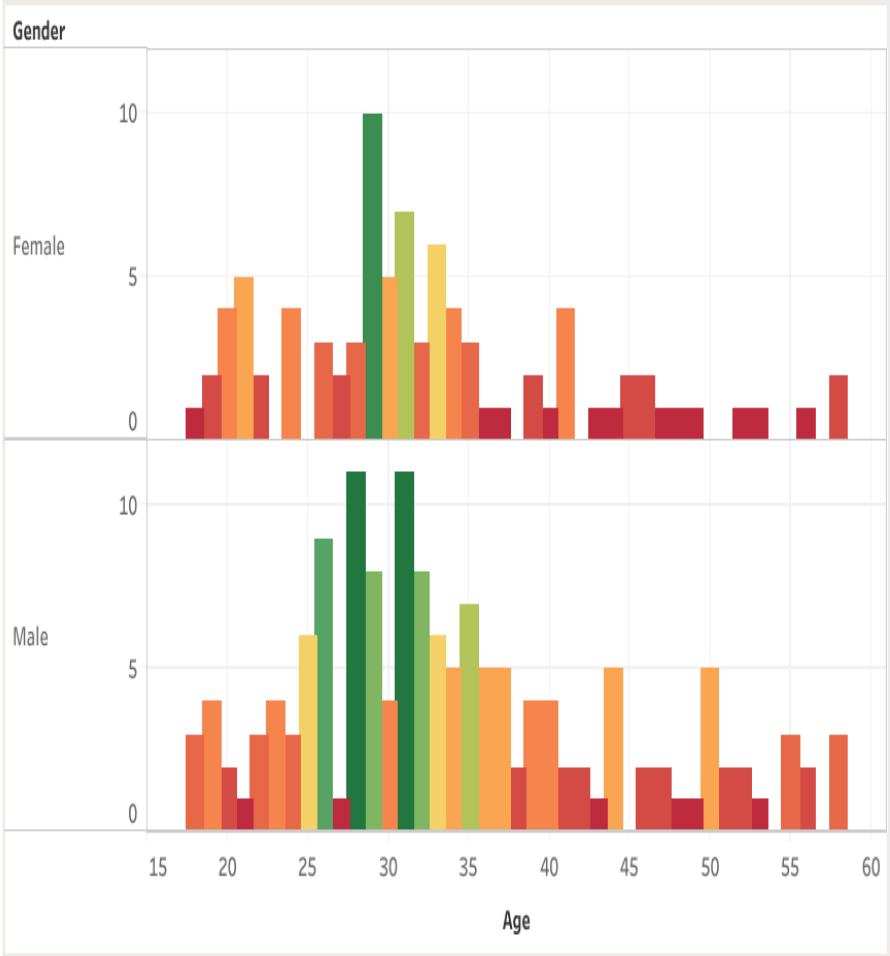
LINK ZU TABLEAU: https://public.tableau.com/app/profile/manuela.holzner/viz/HumanRessource/Dashboard1

Age and Education

Education
Alle

Age Distribution and Attrition

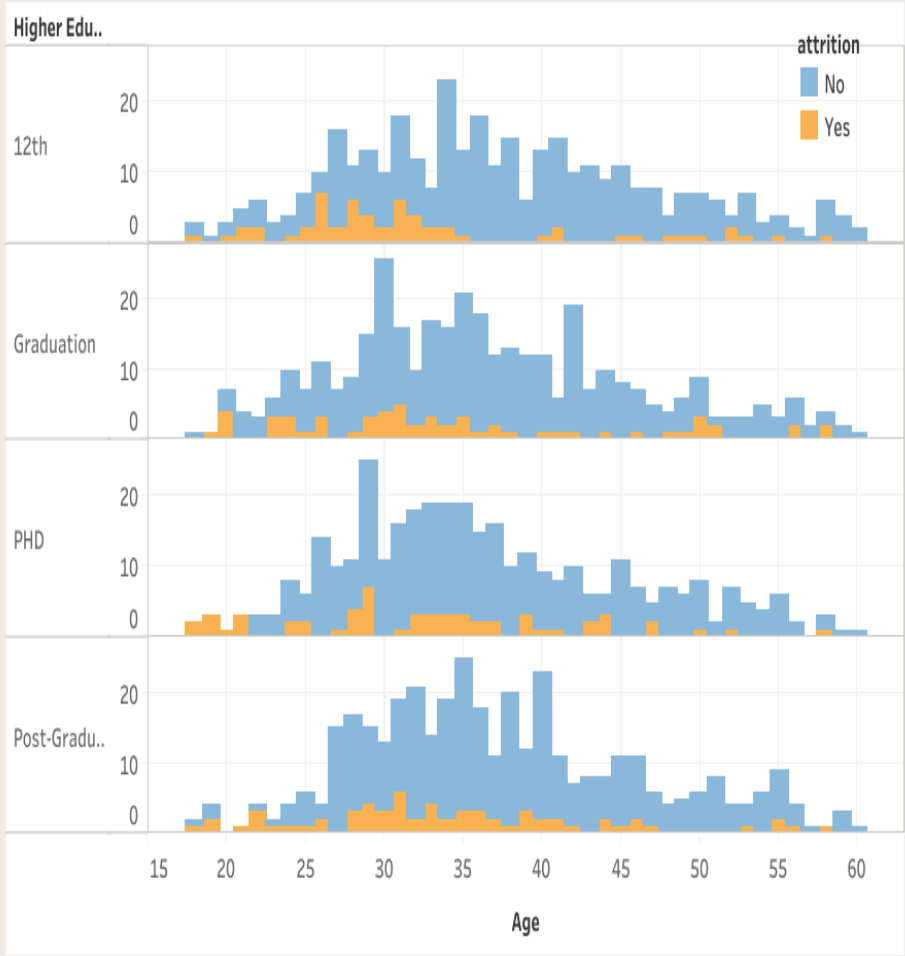
Attrition
Yes



Education and Attrition

Department
Alle

Jobrole
Alle



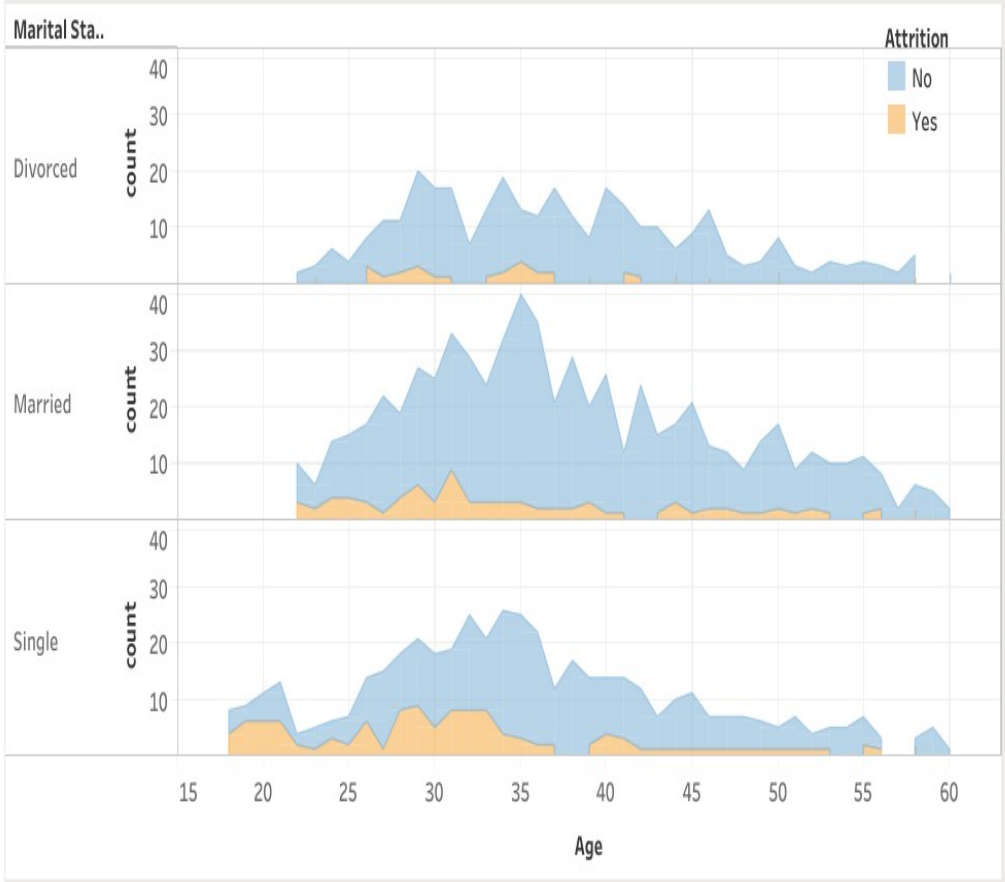
Marital Status and Status of Leaving

Maritalstatus
Alle

Department
Alle

Jobrole
Alle

Marital Status and Attrition



Attrition - Reason

Status of leaving
Alle

Status of le..	Department	Job Role	Attrition	
Better Opportunity	Human Resources	Human Resources	No	5
			Yes	1
		Manager	No	1
	Research & Development	Healthcare Representative	No	22
			Yes	3
		Laboratory Technician	No	41
			Yes	5
		Manager	No	11
		Manufacturing Director	No	38
			Yes	2
		Research Director	No	21
		Research Scientist	No	44
			Yes	6
	Sales	Manager	No	6
			Yes	2
		Sales Executive	No	47
			Yes	14
		Sales Representative	No	7
			Yes	8
Dept.Head	Human Resources	Human Resources	No	8
			Yes	2
		Manager	No	4

Overtime and Business Travel

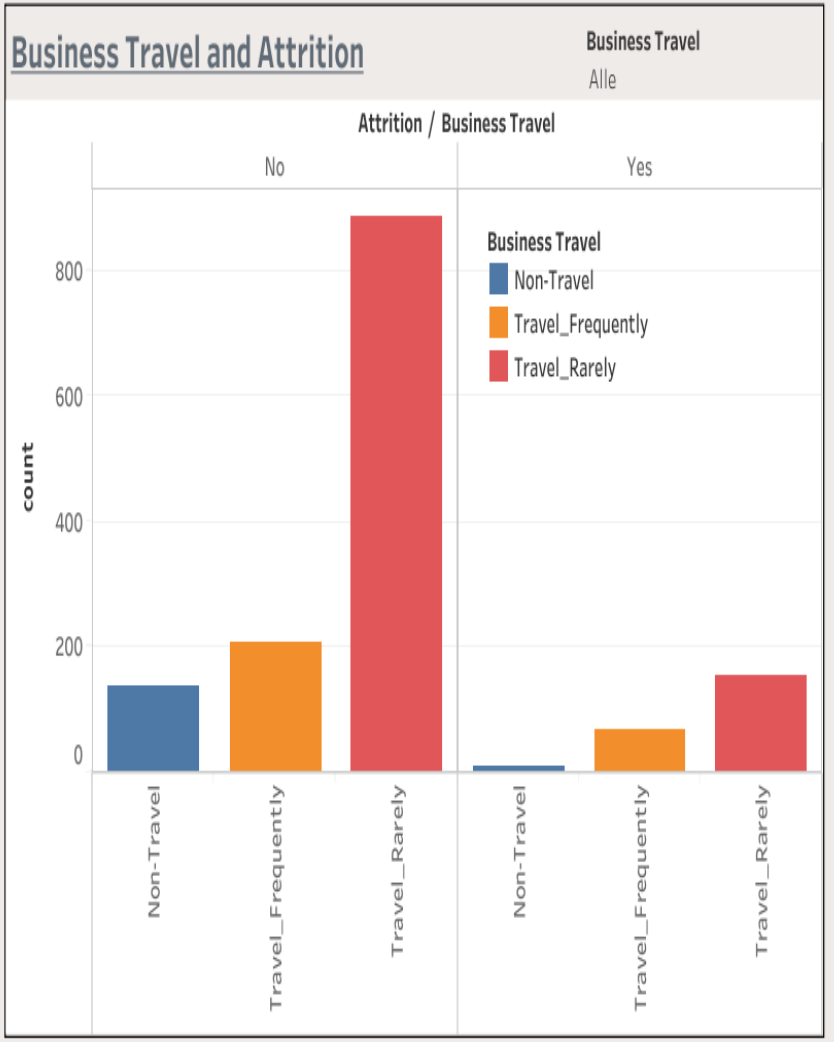
Jobrole
Alle

Overtime and Attrition

		Job Role								
Attrition	Over Time	Healthcare Representative	Human Resources	Laboratory Technician	Manager	Manufacturing Director	Research Director	Research Scientist	Sales Executive	Sales Representative
No	No	87	32	166	74	100	56	181	206	42
	Yes	35	8	31	23	35	22	64	63	8
Yes	Yes	2	5	31	4	4	1	33	31	16
	No	7	7	31	1	6	1	14	26	17

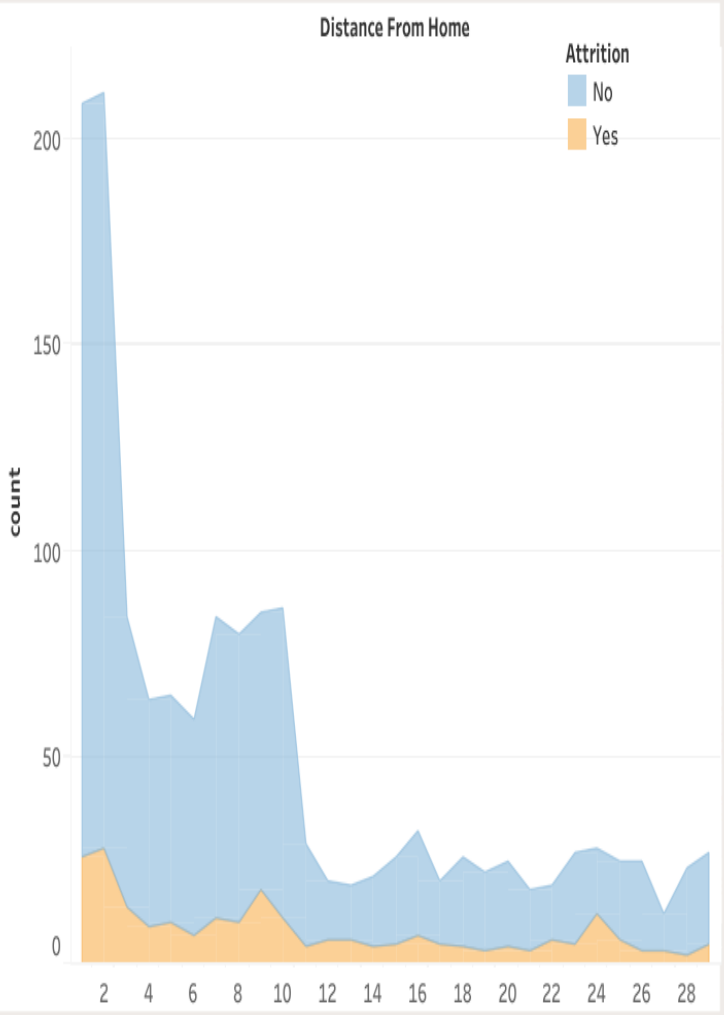
Business Travel and Attrition

Business Travel
Alle

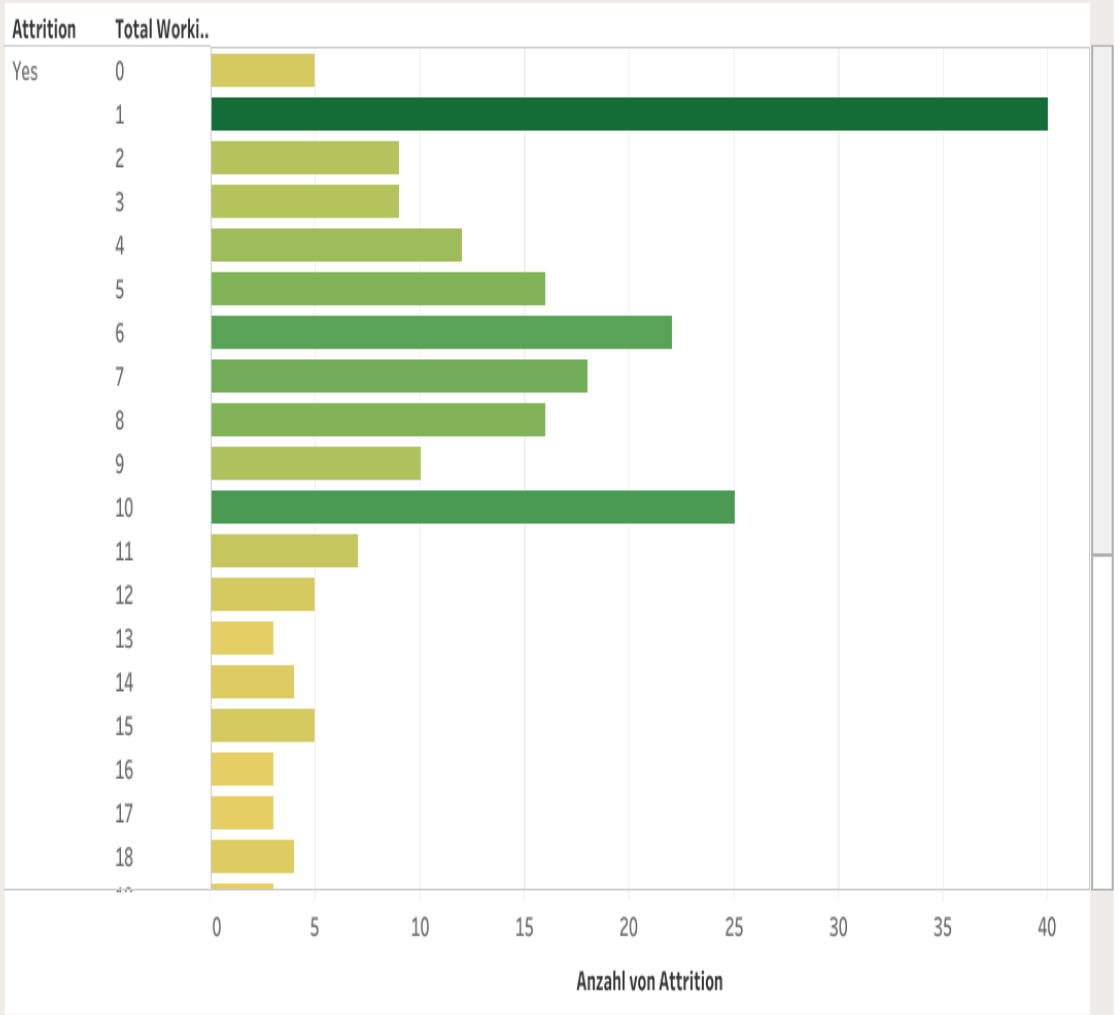


Distance from home and Total working years

Distance from home and Attrition



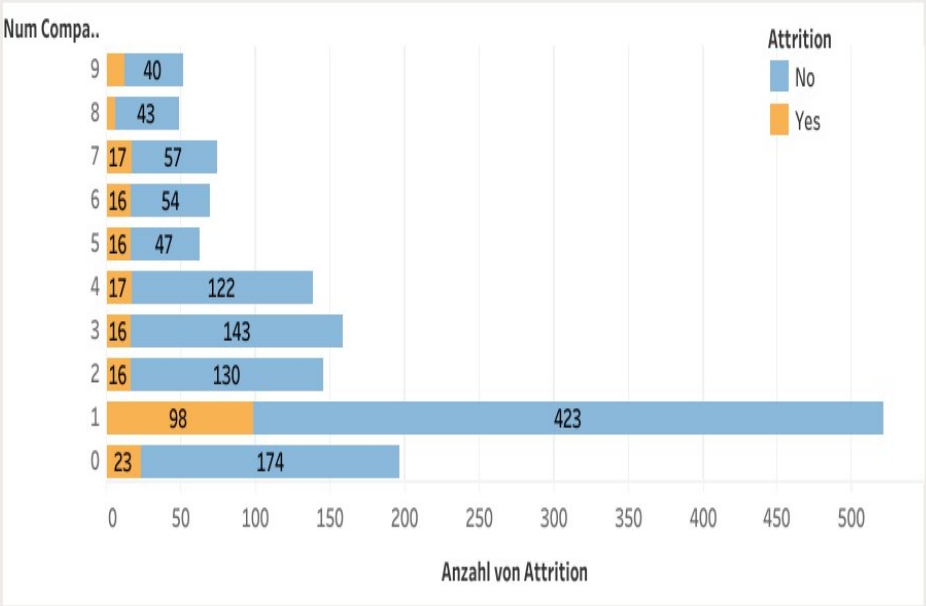
Total working years and Attrition



Other features

Monthly Income
1009 bis 19999

Different companys and Attrition



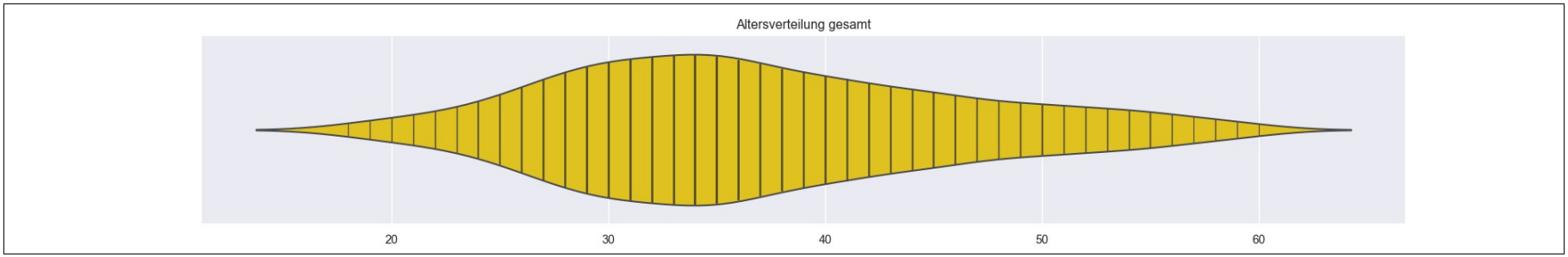
Mode of work/Source of Hire and Attrition

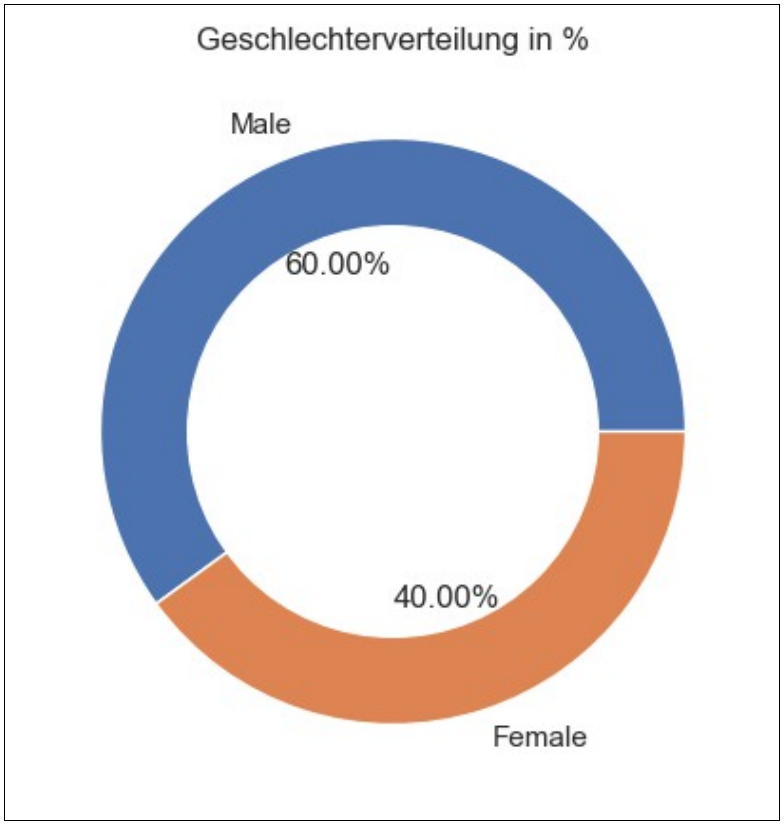
		Attrition	
Mode of wo..	Source of H..	No	Yes
OFFICE	Job Event	124	29
	Job Portal	159	25
	Recruiter	163	34
	Walk-in	141	27
WFH	Job Event	191	28
	Job Portal	129	34
	Recruiter	164	29
	Walk-in	162	31

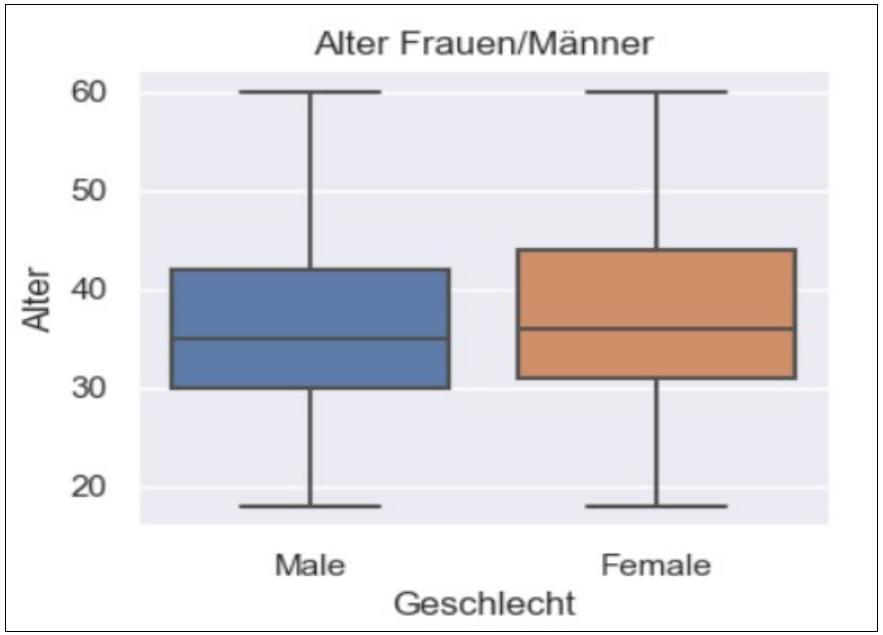
Work accident/Job satisfaction

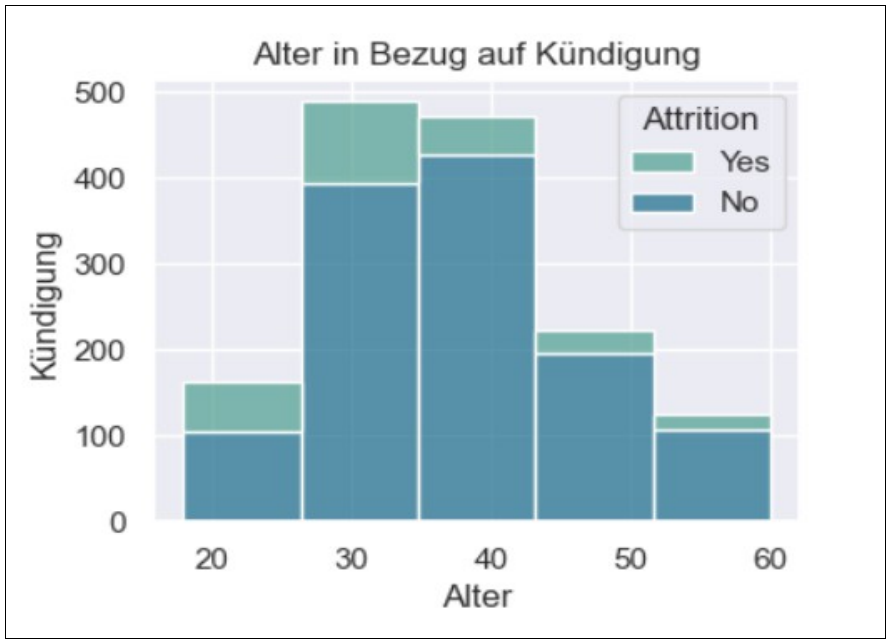
		Attrition	
Work accid..	Job Satisfa..	No	Yes
No	1	112	28
	2	117	23
	3	183	42
	4	208	23
Yes	1	111	38
	2	117	23
	3	186	31
	4	199	29

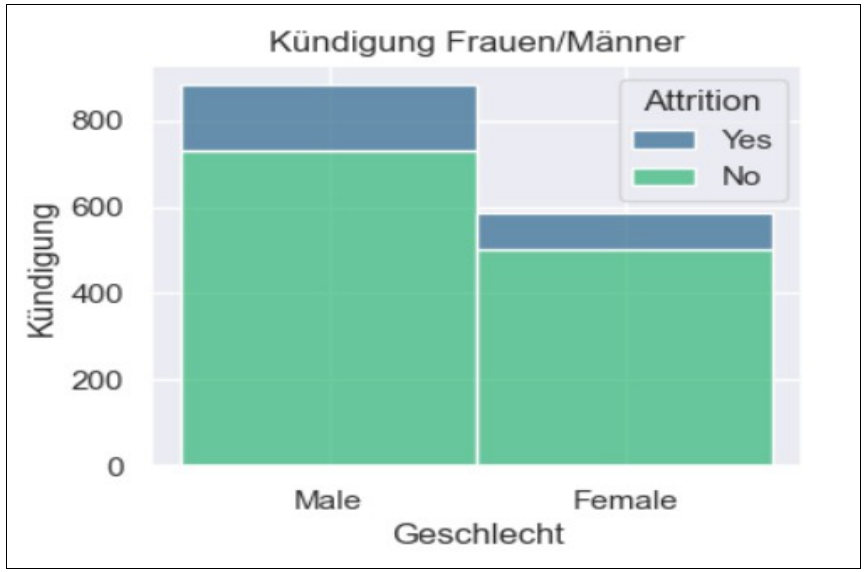
VISUALISIERUNG DES DATENSATZES JUPYTER NOTEBOOK:

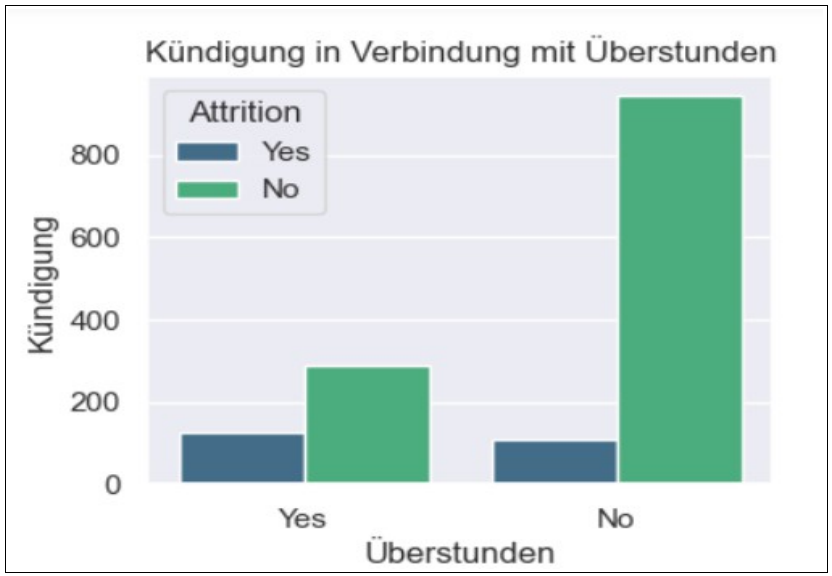


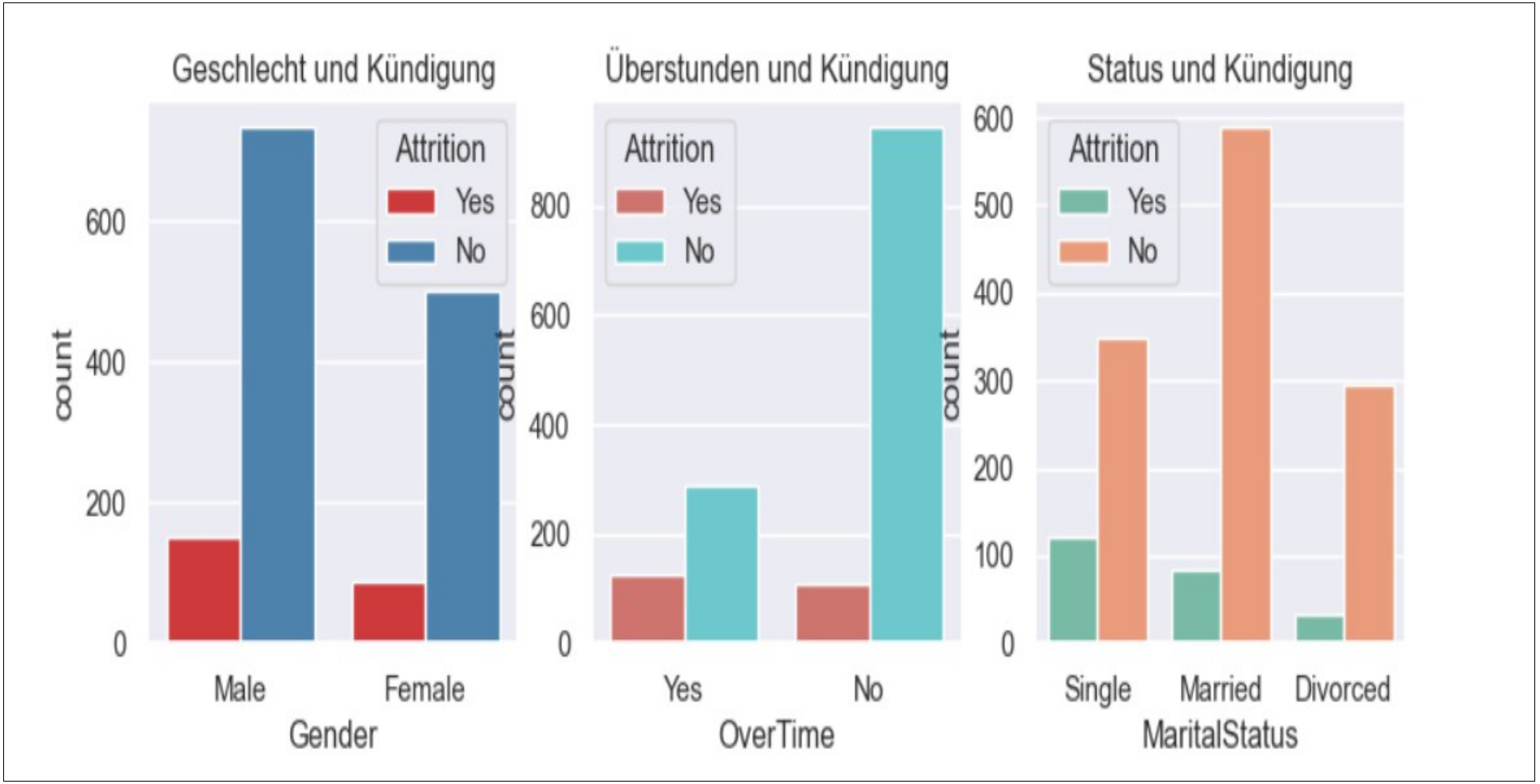


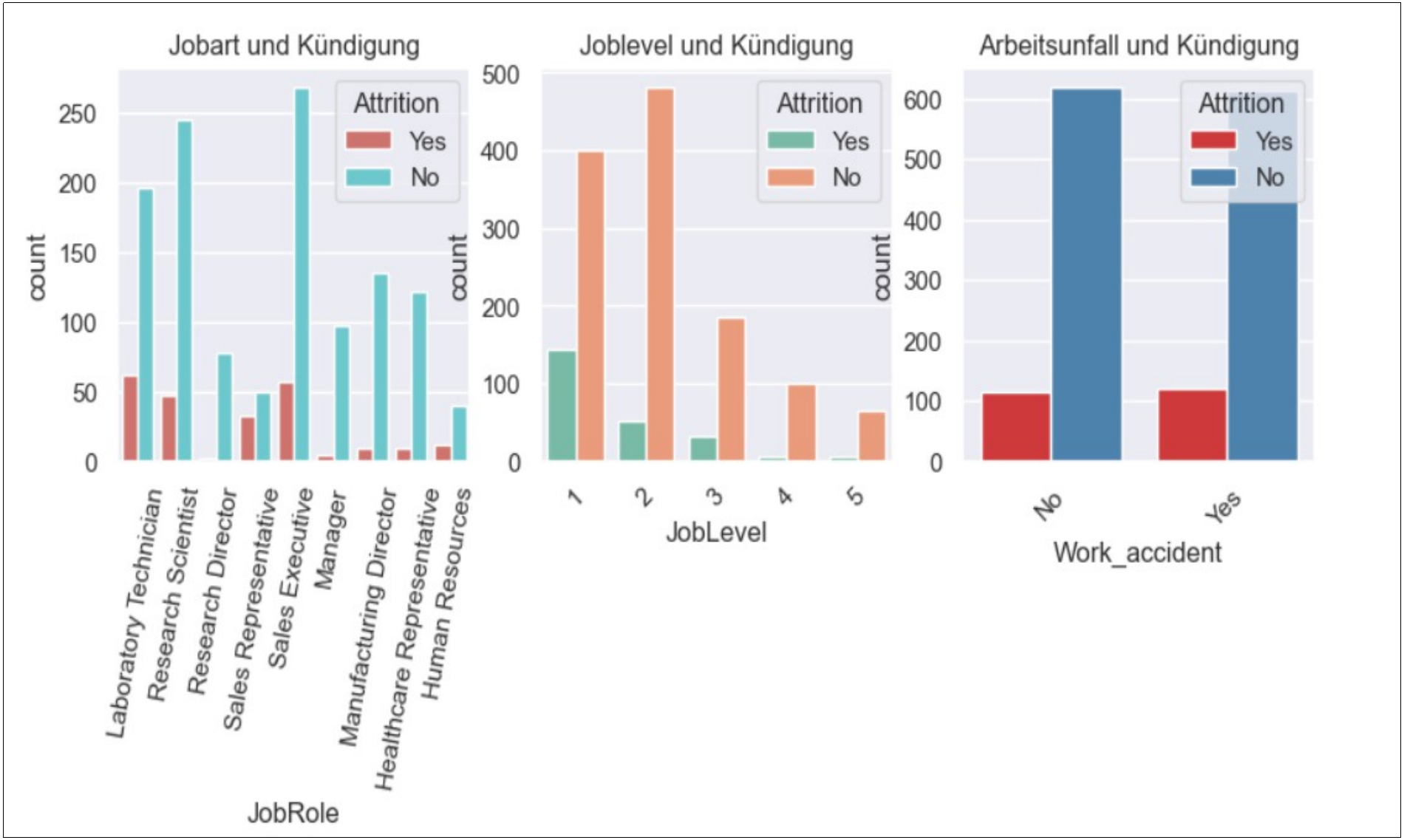

Alter Frauen/Männer
60
50
40
30
20
Alter
Male
Female
Geschlecht

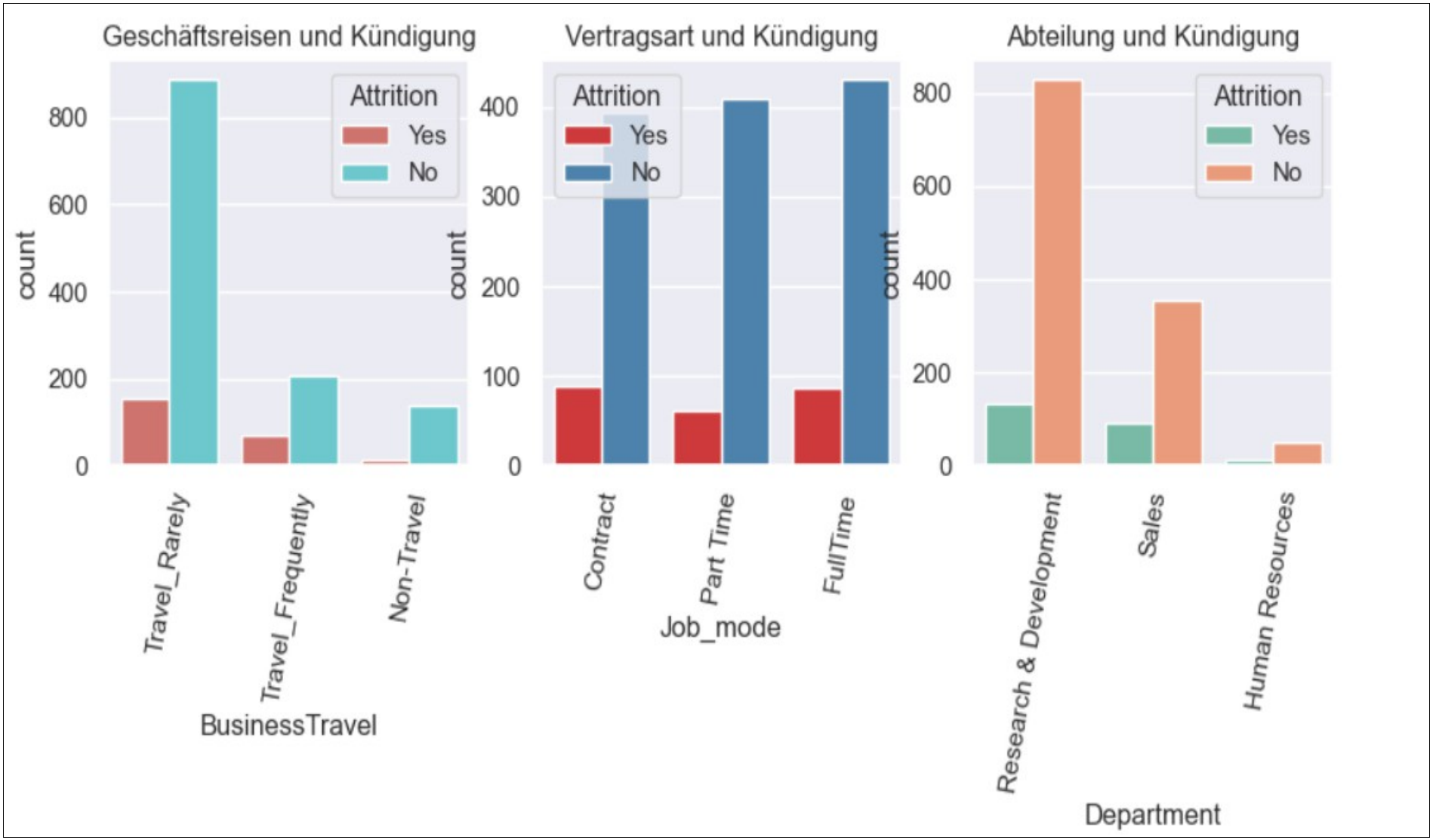

Alter in Bezug auf Kündigung
500
400
300
200
100
0
Kündigung
Attrition
Yes
No
20
30
40
50
60
Alter


Kündigung Frauen/Männer
Attrition
Yes
No
800
600
400
200
0
Kündigung
Male
Female
Geschlecht


Kündigung in Verbindung mit Überstunden
Attrition
Yes
No
800
600
400
200
0
Kündigung
Yes
No
Überstunden


Geschlecht und Kündigung
Attrition
Yes
No
count
0
200
400
600
Male
Female
Gender
Überstunden und Kündigung
Attrition
Yes
No
count
0
200
400
600
800
Yes
No
OverTime
Status und Kündigung
Attrition
Yes
No
count
0
100
200
300
400
500
600
Single
Married
Divorced
MaritalStatus


Jobart und Kündigung
Attrition
Yes
No
count
0
50
100
150
200
250
Laboratory Technician
Research Scientist
Research Director
Sales Representative
Sales Executive
Manager
Manufacturing Director
Healthcare Representative
Human Resources
JobRole
Joblevel und Kündigung
Attrition
Yes
No
count
0
100
200
300
400
500
1
2
3
4
5
JobLevel
Arbeitsunfall und Kündigung
Attrition
Yes
No
count
0
100
200
300
400
500
600
No
Yes
Work_accident


Geschäftsreisen und Kündigung
Attrition
Yes
No
count
800
600
400
200
0
Travel_Rarely
Travel_Frequently
Non-Travel
BusinessTravel
Vertragsart und Kündigung
Attrition
Yes
No
count
400
300
200
100
0
Contract
Part Time
FullTime
Job_mode
Abteilung und Kündigung
Attrition
Yes
No
count
800
600
400
200
0
Research & Development
Sales
Human Resources
Department

SUPERVISED LEARNING:

- ✗ Verwendete Algorithmen:

```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier

log = LogisticRegression()
knc = KNeighborsClassifier()
svc = SVC()
nab = GaussianNB()
rfc = RandomForestClassifier()
```

- ✗ Standardscaler für Streuung und Varianz und train_test_split mit test_size 0.20:

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X01, y01, test_size = 0.20, random_state = 33)
```

✗ Unterteilung Durchläufe:

alle features (Variable X01) miteinbeziehen

```
X01 = mitarbeiter02.drop(['Attrition'], axis=1)
y01 = mitarbeiter02['Attrition']
```

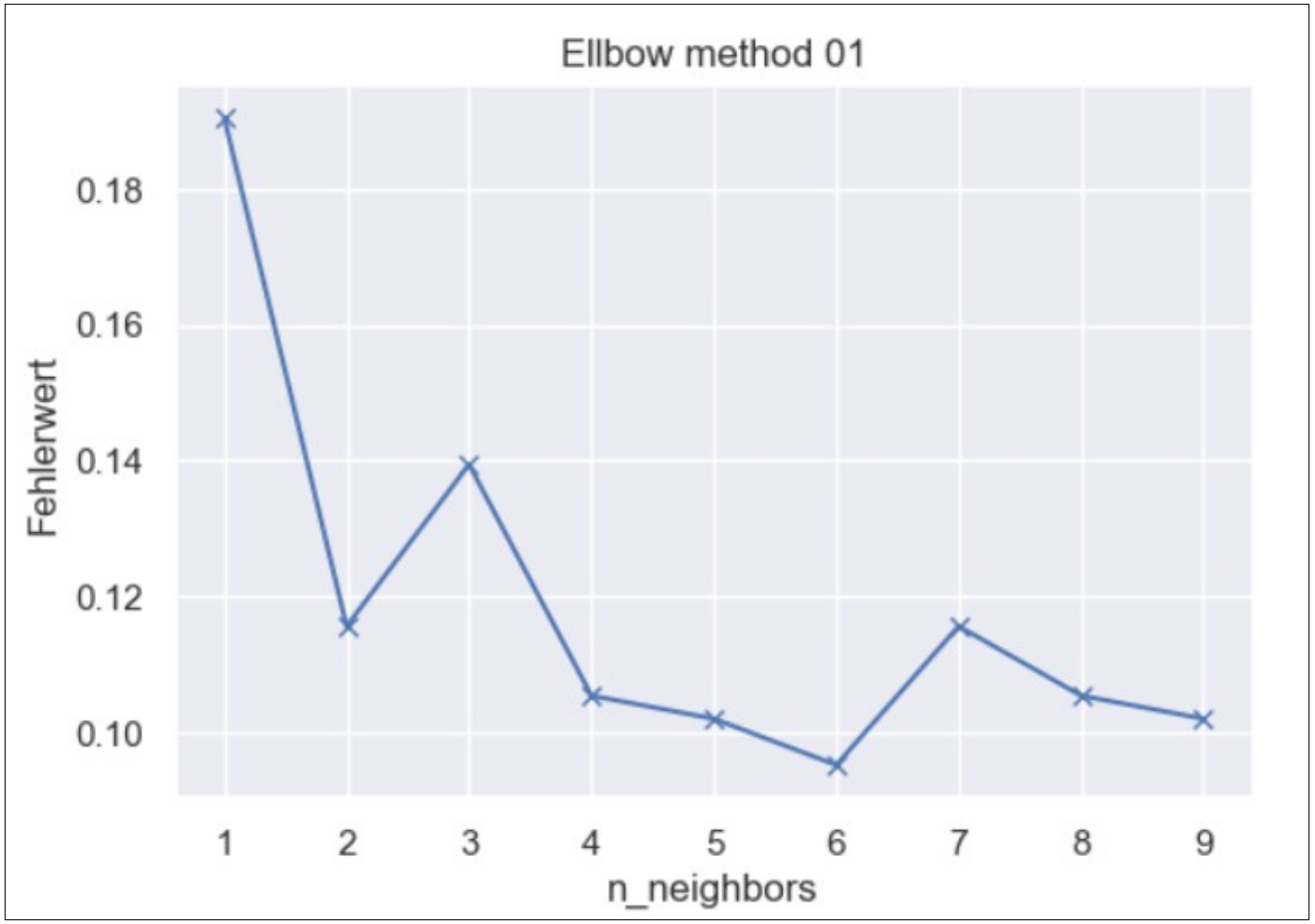
nur features mit höchsten Korrelationen (Variable X02)

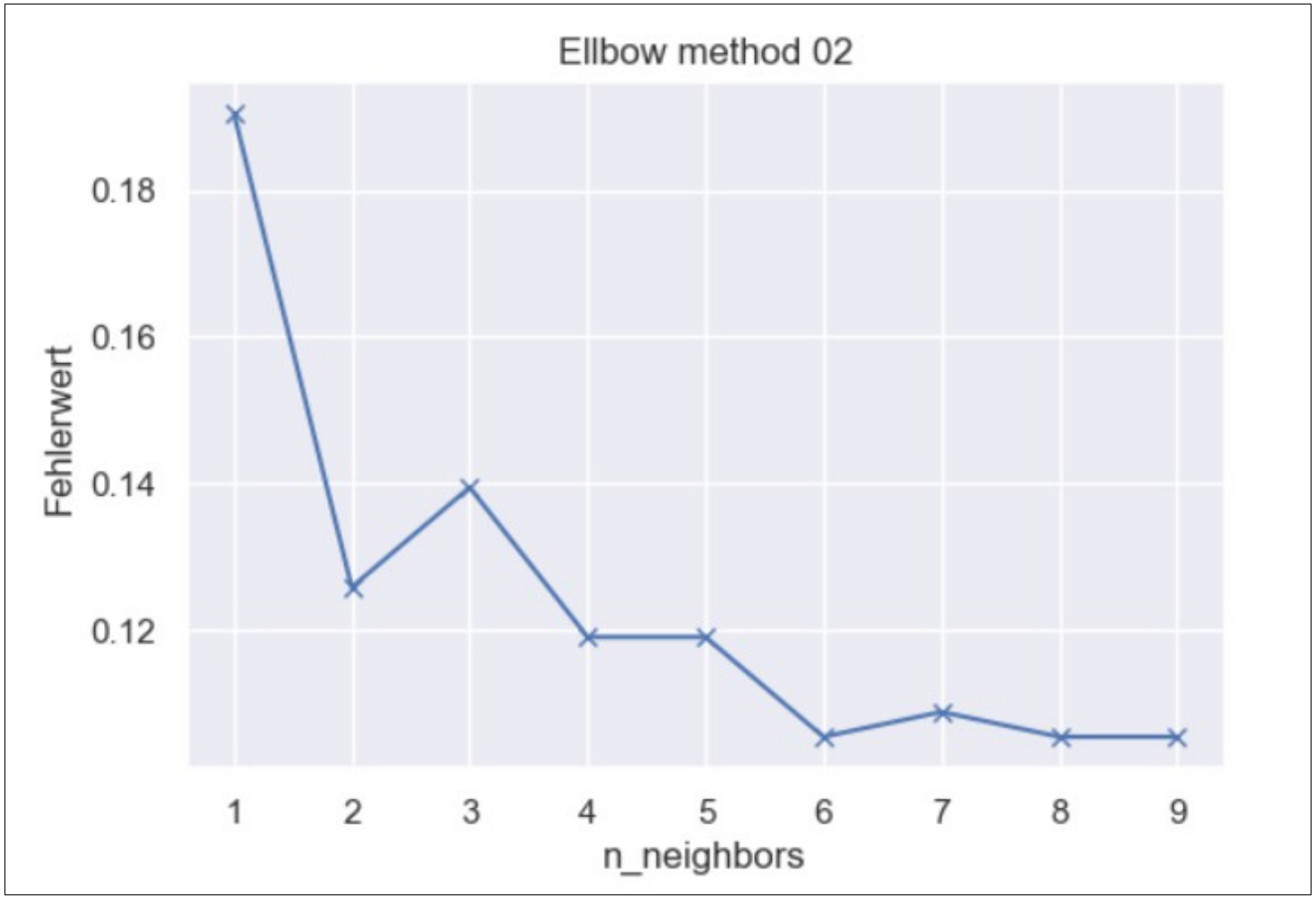
```
X03 = mitarbeiter02[["DistanceFromHome", "NumCompaniesWorked", "Gender", "Source_of_Hire", "Work_accident",
                     "Mode_of_work", "Status_of_leaving", "Higher_Education", "PerformanceRating", "PercentSalaryHike",
                     "YearsSinceLastPromotion", "JobRole", "Leaves", "Absenteeism", "Job_mode",
                     "TrainingTimesLastYear", "Department"]]
y03 = mitarbeiter02["Attrition"]
```

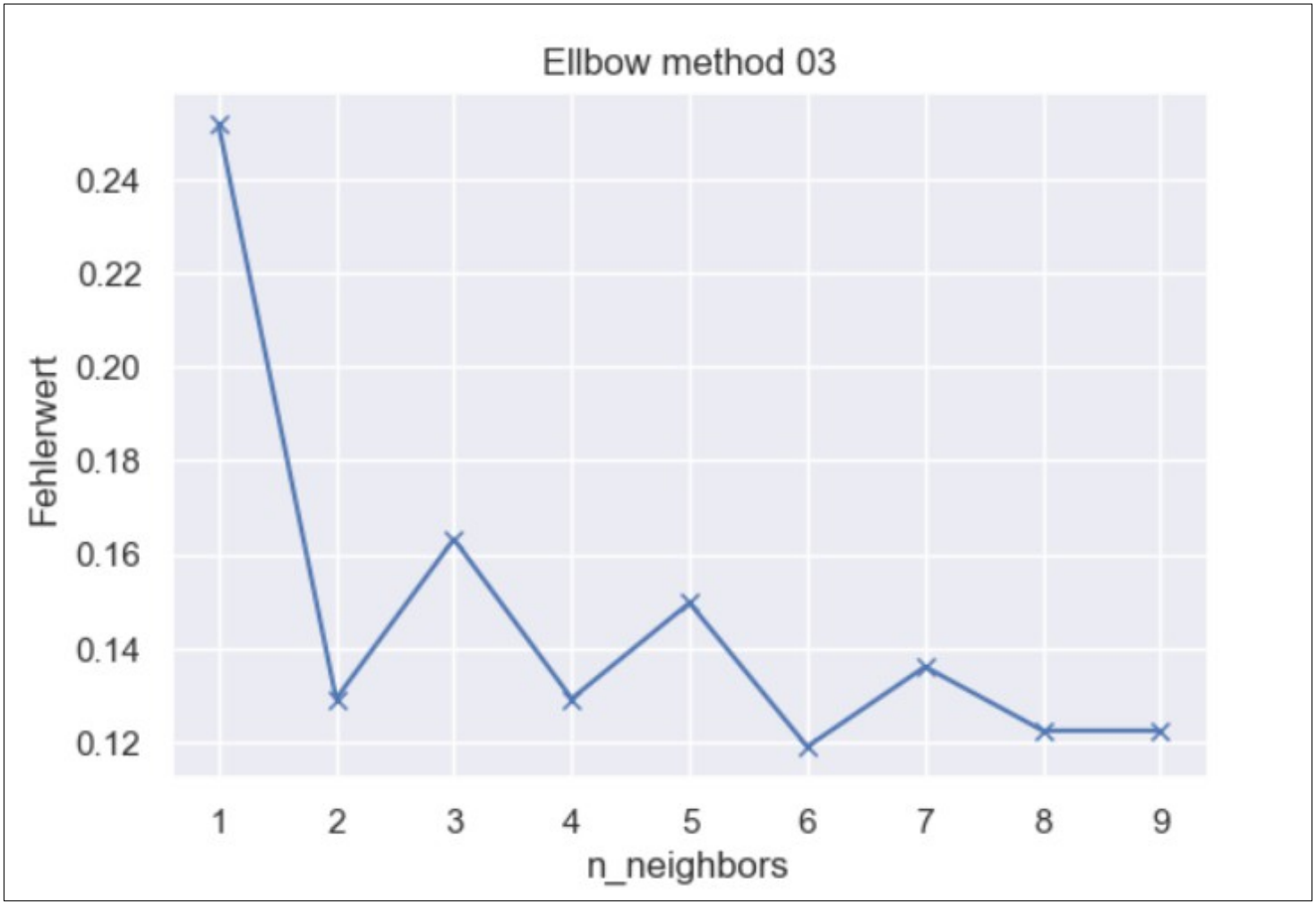
nur features mit niedrigsten Korrelationen (Variable X03)

```
X02 = mitarbeiter02[["OverTime", "BusinessTravel", "JobSatisfaction", "YearsAtCompany", "JobInvolvement",
                     "MaritalStatus", "YearsWithCurrManager", "MonthlyIncome", "Age", "TotalWorkingYears", "JobLevel"]]
y02 = mitarbeiter02["Attrition"]
```

✗ Ellbow-methods für KNN X01/X02/X03 – mit 6 neighbors von KNN bestes Ergebnis:




Ellbow method 02
Fehlerwert
0.18
0.16
0.14
0.12
1
2
3
4
5
6
7
8
9
n_neighbors


Ellbow method 03
Fehlerwert
0.24
0.22
0.20
0.18
0.16
0.14
0.12
1
2
3
4
5
6
7
8
9
n_neighbors

✗ Gridsearch für beste Parameter von SVC für X01/X02/X03:

```
from sklearn.model_selection import GridSearchCV

# dictionary mit Werten für C und gamma
hyperparameter = {"C" : [0.1, 1, 10, 100, 1000], "gamma" : [1, 0.1, 0.01, 0.001, 0.0001]}
```

✗ unterschiedliche Werte erhalten:

X01:

```
1 grid01.best_params_
```

```
{'C': 1000, 'gamma': 0.0001}
```

X02:

```
1 grid02.best_params_
```

```
{'C': 10, 'gamma': 0.01}
```

X03:

```
1 grid03.best_params_
```

```
{'C': 0.1, 'gamma': 1}
```

✗ mit allen Algorithmen trainieren, vorhersagen und reports:

```
TRAINIEREN

log = LogisticRegression().fit(X_train, y_train)
knc = KNeighborsClassifier(n_neighbors = 6).fit(X_train, y_train)
svc = SVC(C=1000, gamma=0.0001).fit(X_train, y_train)
nab = GaussianNB().fit(X_train, y_train)
rfc = RandomForestClassifier(random_state = 33, n_estimators = 1000).fit(X_train, y_train)
```

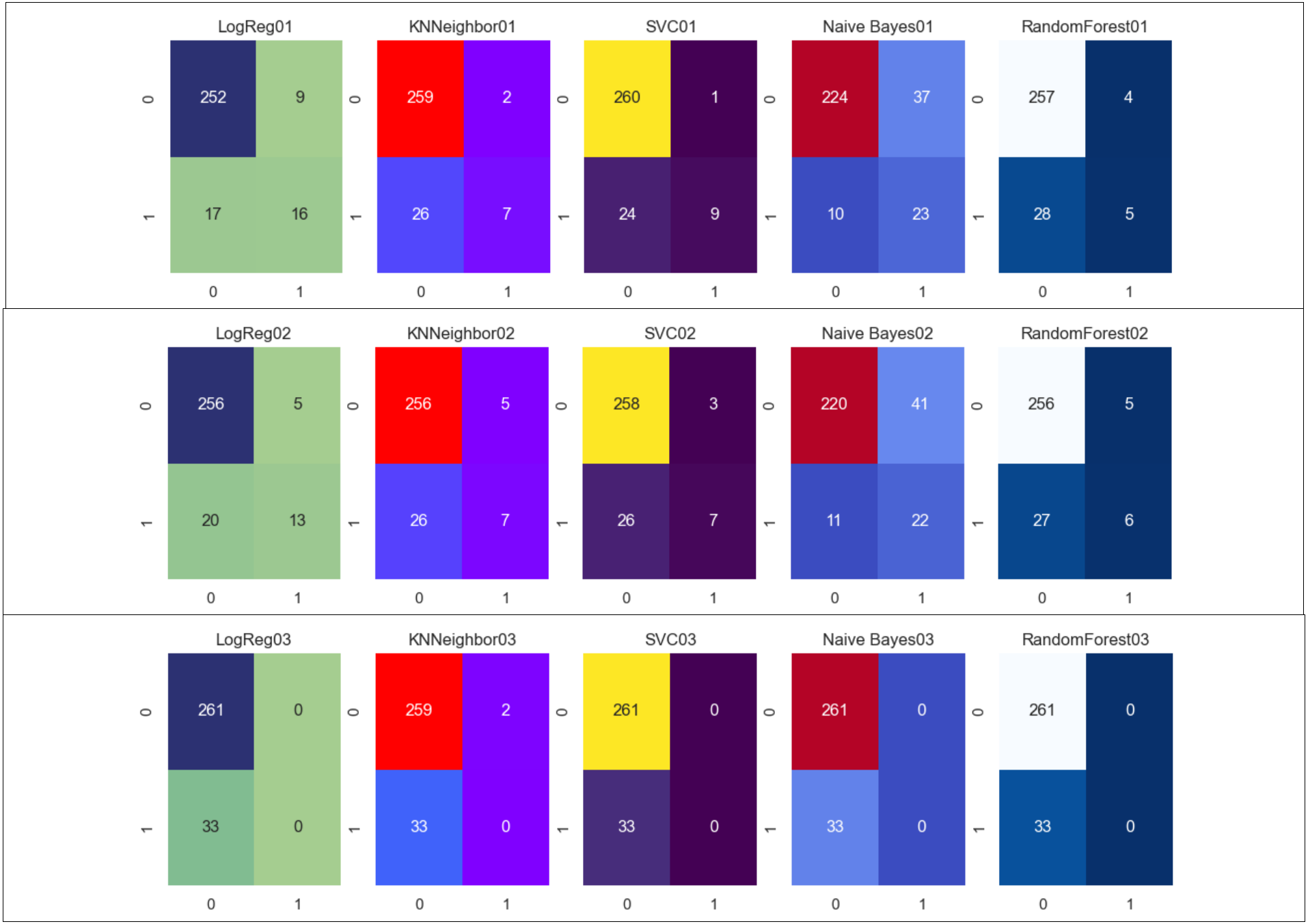
```
VORHERSAGEN

pred_log = log.predict(X_test)
pred_knc = knc.predict(X_test)
pred_svc = svc.predict(X_test)
pred_nab = nab.predict(X_test)
pred_rfc = rfc.predict(X_test)
```

```
REPORTS

print("Genauigkeit LogReg: {:.2f}%".format((accuracy_score(y_test, pred_log)*100)))
print("Genauigkeit KNN: {:.2f}%".format((accuracy_score(y_test, pred_knc)*100)))
print("Genauigkeit SVC: {:.2f}%".format((accuracy_score(y_test, pred_svc)*100)))
print("Genauigkeit NAIVE: {:.2f}%".format((accuracy_score(y_test, pred_nab)*100)))
print("Genauigkeit Random: {:.2f}%".format((accuracy_score(y_test, pred_rfc)*100)))
```

✗ Reports:

```
ALLE FEATURES:

Genauigkeit LogReg01:      91.16%
Genauigkeit KNN01   :      90.48%
Genauigkeit SVC01   :      91.50%
Genauigkeit NAIVE01 :      84.01%
Genauigkeit Random01:      89.12%

FEATURES MIT HÖCHSTEN KORRELATIONEN:

Genauigkeit LogReg02:      91.50%
Genauigkeit KNN02   :      89.46%
Genauigkeit SVC02   :      90.14%
Genauigkeit NAIVE02 :      82.31%
Genauigkeit Random02:      89.12%

FEATURES MIT GERINGSTEN KORRELATIONEN:

Genauigkeit LogReg03:      88.78%
Genauigkeit KNN03   :      88.10%
Genauigkeit SVC03   :      88.78%
Genauigkeit NAIVE03 :      88.78%
Genauigkeit Random03:      88.78%
```

✗ Confusion matrix:



UNSUPERVISED LEARNING:

- ✗ Verwendete Algorithmen:

```
PCA:

pca = PCA(n_components=1, random_state=33)
```

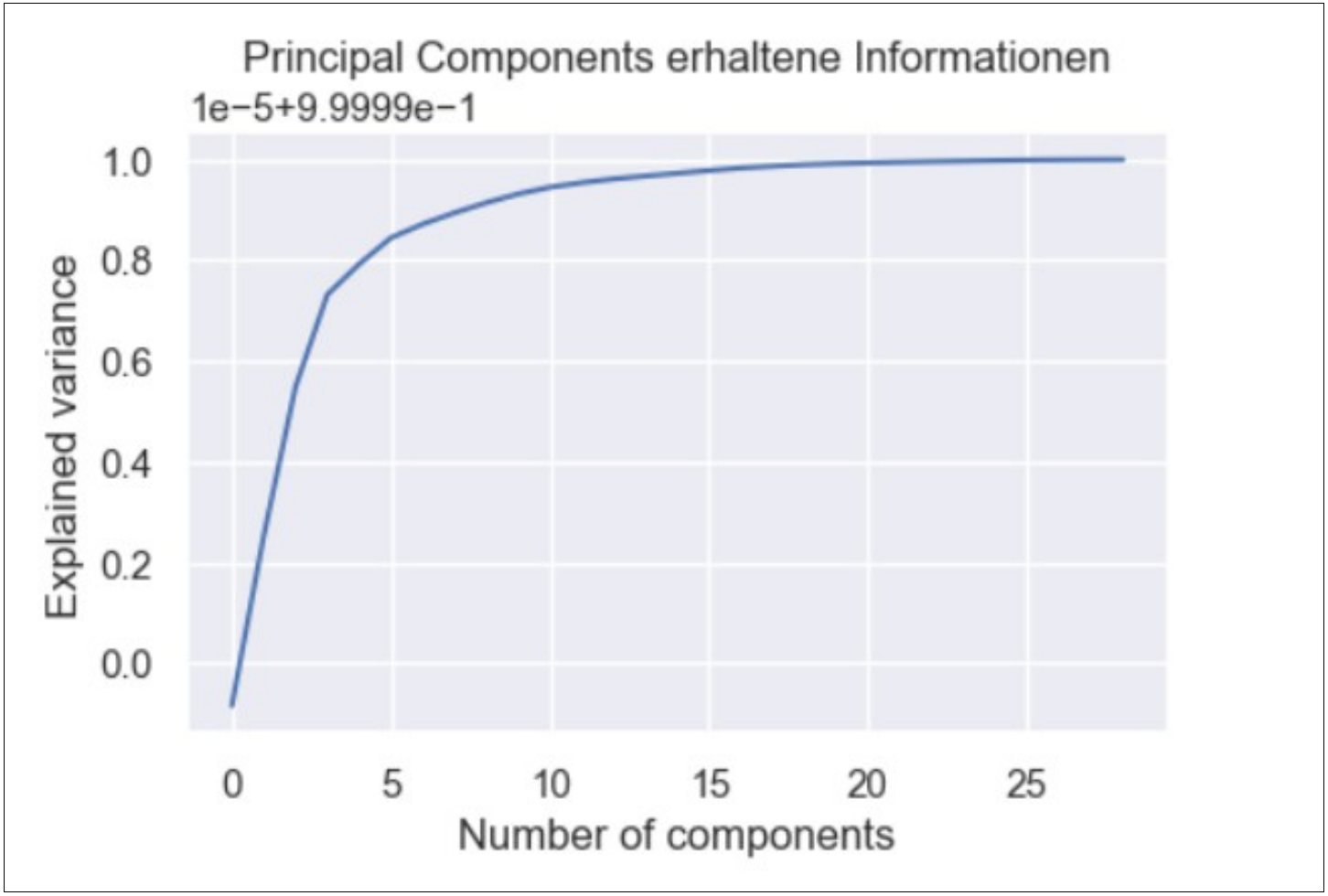
```
Algorithmen, die nach der Reduzierung mit PCA verwendet wurden:

log04 = LogisticRegression()
knc04 = KNeighborsClassifier()
svc04 = SVC()
nab04 = GaussianNB()
rfc04 = RandomForestClassifier()
```

```
K-Means:

km = KMeans(n_clusters=2, random_state=33)
```

- ✗ Standardscaler für Streuung und Varianz:

```
Standardscaler:

scaled_X04 = scaler.fit_transform(X04)
```

- PCA – Überprüfung der enthaltenen Informationen in Abhängigkeit der Principal Components :



✗ Erste Komponente enthält nahezu 100 % der Informationen → n_components=1:

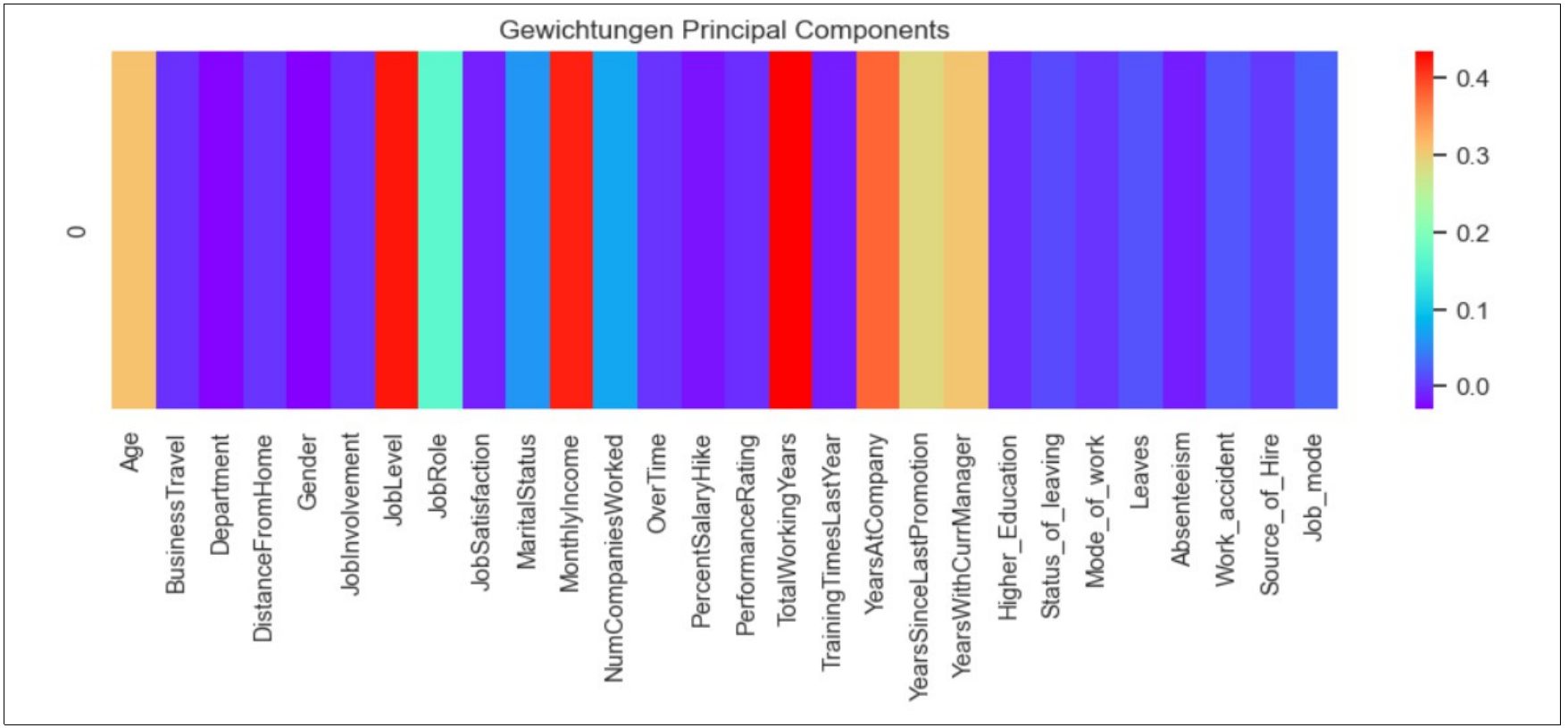
```
2
3  for i,value in enumerate(pca_check.explained_variance_ratio_):
4      print(f"{i+1}. Principal Component erklärt {value*100:.4f}% der Varianz ")
```

```
1. Principal Component erklärt 99.9989% der Varianz
2. Principal Component erklärt 0.0003% der Varianz
3. Principal Component erklärt 0.0003% der Varianz
4. Principal Component erklärt 0.0002% der Varianz
5. Principal Component erklärt 0.0001% der Varianz
6. Principal Component erklärt 0.0001% der Varianz
7. Principal Component erklärt 0.0000% der Varianz
8. Principal Component erklärt 0.0000% der Varianz
9. Principal Component erklärt 0.0000% der Varianz
```

✗ Trainieren mit PCA 1 Dimension:

```
PCA: trainieren und transformieren:

x_pca = pca.fit_transform(scaled_X04)
```

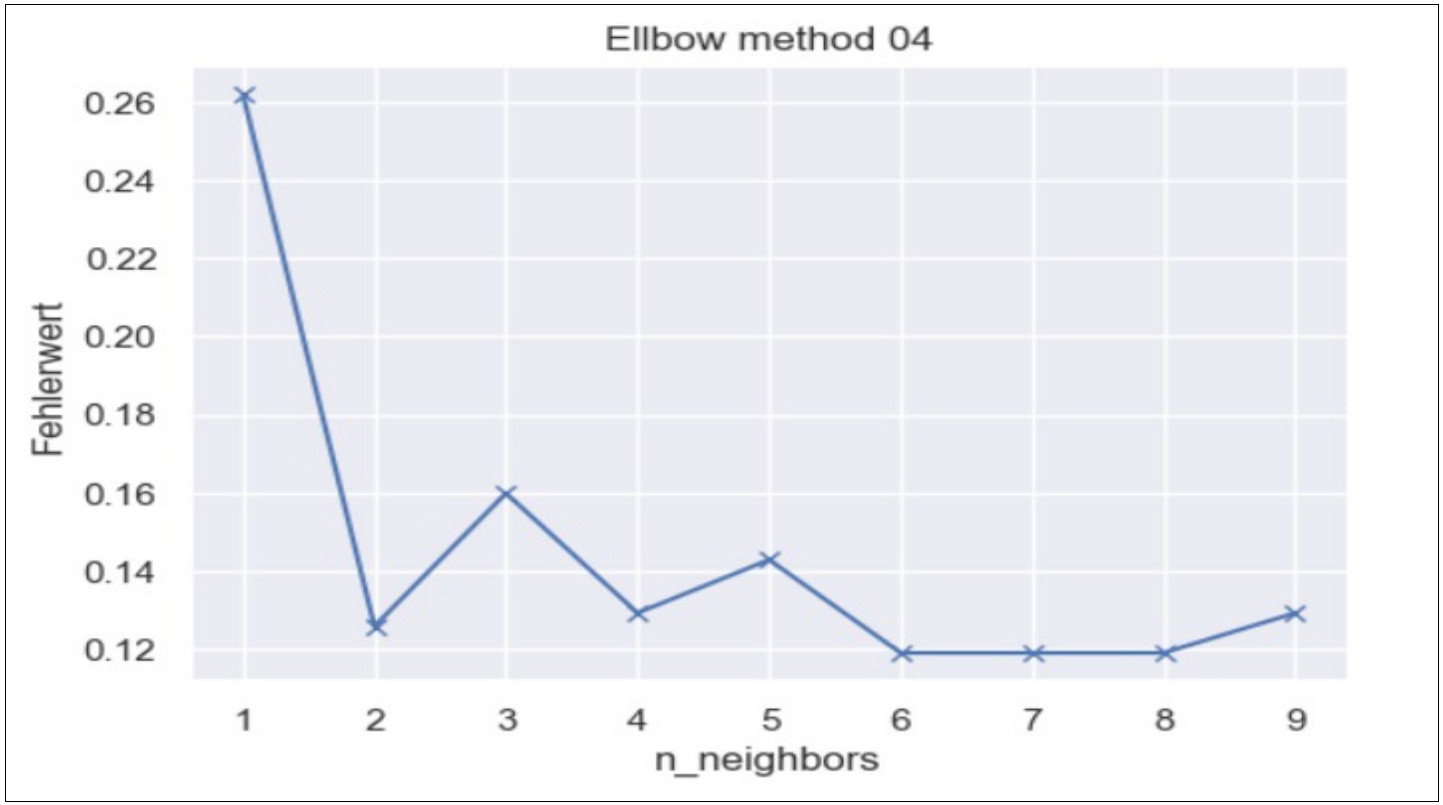
✗ Gewichtung Principal Components mit den maximal enthaltenen Informationen:



✗ SUPERVISED LEARNING ALGORITHMEN MIT PCA REDUZIERTEN FEATURES:

```
X04 = x_pca
y04 = mitarbeiter02["Attrition"]
```

✗ Ellbow-method für KNN X04:



✗ Gridsearch für beste Parameter von SVC:

```
hyperparameter = {"C" : [0.1, 1, 10, 100, 1000], "gamma" : [1, 0.1, 0.01, 0.001, 0.0001]}

grid04 = GridSearchCV(SVC(), hyperparameter, refit = True)
grid04.fit(X_train04, y_train04)
```

X04

```
1  grid04.best_params_
```

```
{'C': 100, 'gamma': 1}
```

✗ mit allen Algorithmen trainieren und vorhersagen:

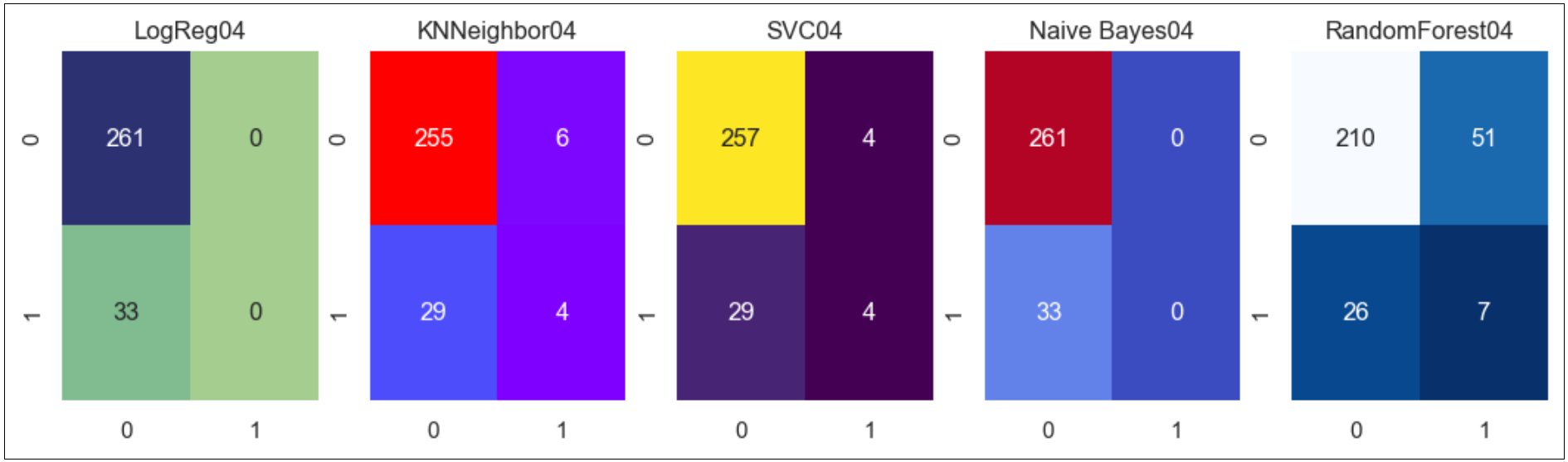
```
TRAINIEREN

log04 = LogisticRegression().fit(X_train04, y_train04)
knc04 = KNeighborsClassifier(n_neighbors = 6).fit(X_train04, y_train04)
svc04 = SVC(C=100, gamma=1).fit(X_train04, y_train04)
nab04 = GaussianNB().fit(X_train04, y_train04)
rfc04 = RandomForestClassifier(random_state = 33, n_estimators = 1000).fit(X_train04, y_train04)
```

```
VORHERSAGEN

pred_log04 = log04.predict(X_test04)
pred_knc04 = knc04.predict(X_test04)
pred_svc04 = svc04.predict(X_test04)
pred_nab04 = nab04.predict(X_test04)
pred_rfc04 = rfc04.predict(X_test04)
```

✗ Reports gesamt:

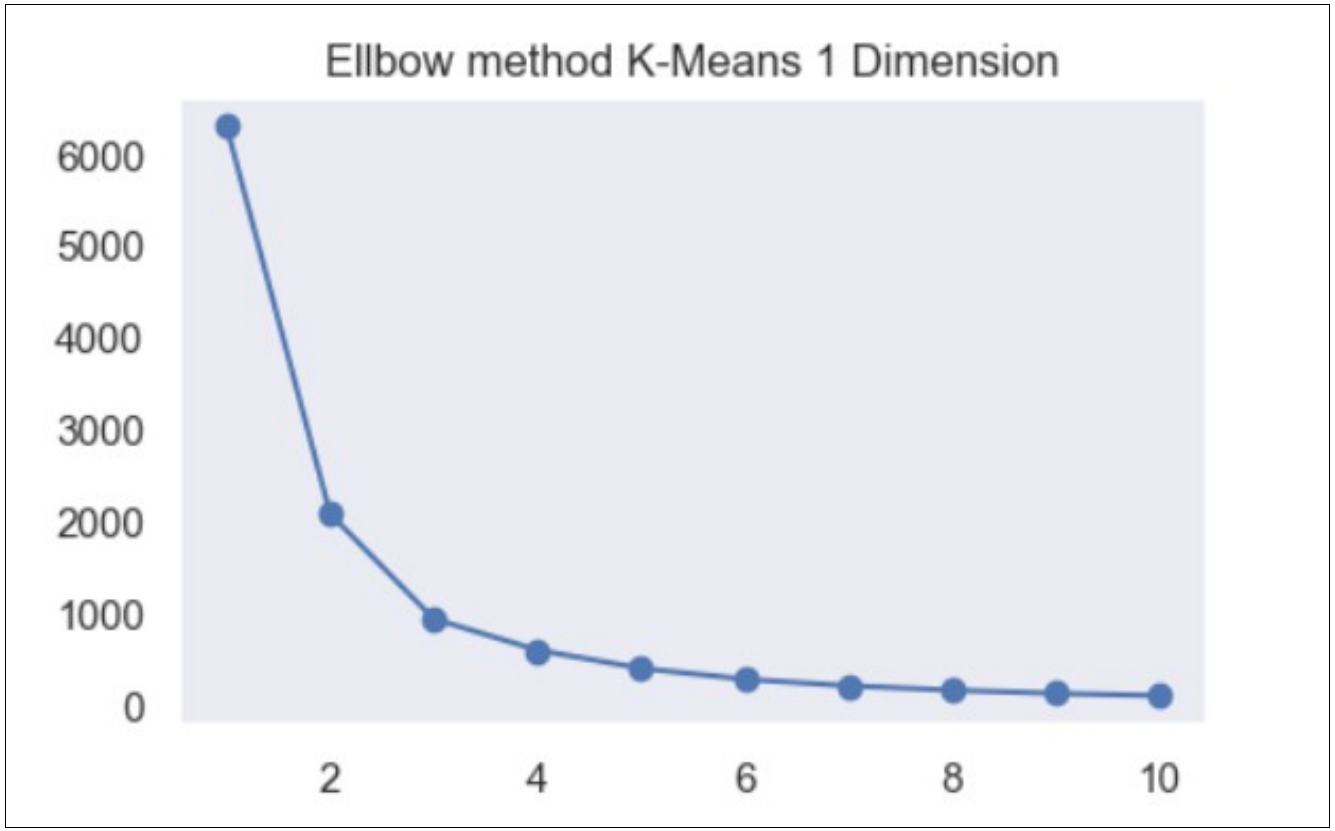
```
ALLE FEATURES:

Genauigkeit LogReg01:       91.16%
Genauigkeit KNN01    :      90.48%
Genauigkeit SVC01    :      91.50%
Genauigkeit NAIVE01  :      84.01%
Genauigkeit Random01:       89.12%

FEATURES MIT HÖCHSTEN KORRELATIONEN:

Genauigkeit LogReg02:       91.50%
Genauigkeit KNN02    :      89.46%
Genauigkeit SVC02    :      90.14%
Genauigkeit NAIVE02  :      82.31%
Genauigkeit Random02:       89.12%

FEATURES MIT GERINGSTEN KORRELATIONEN:

Genauigkeit LogReg03:       88.78%
Genauigkeit KNN03    :      88.10%
Genauigkeit SVC03    :      88.78%
Genauigkeit NAIVE03  :      88.78%
Genauigkeit Random03:       88.78%

FEATURES MIT PCA REDUZIERT:

Genauigkeit LogReg04:       88.78%
Genauigkeit KNN04    :      88.10%
Genauigkeit SVC04    :      88.78%
Genauigkeit NAIVE04  :      88.78%
Genauigkeit Random04:       73.81%
```

✗ Confusion matrix mit PCA:



✗ K-MEANS MIT PCA

✗ Ellbow-method für K-Means:



✗ mit K-Means trainieren und vorhersagen:

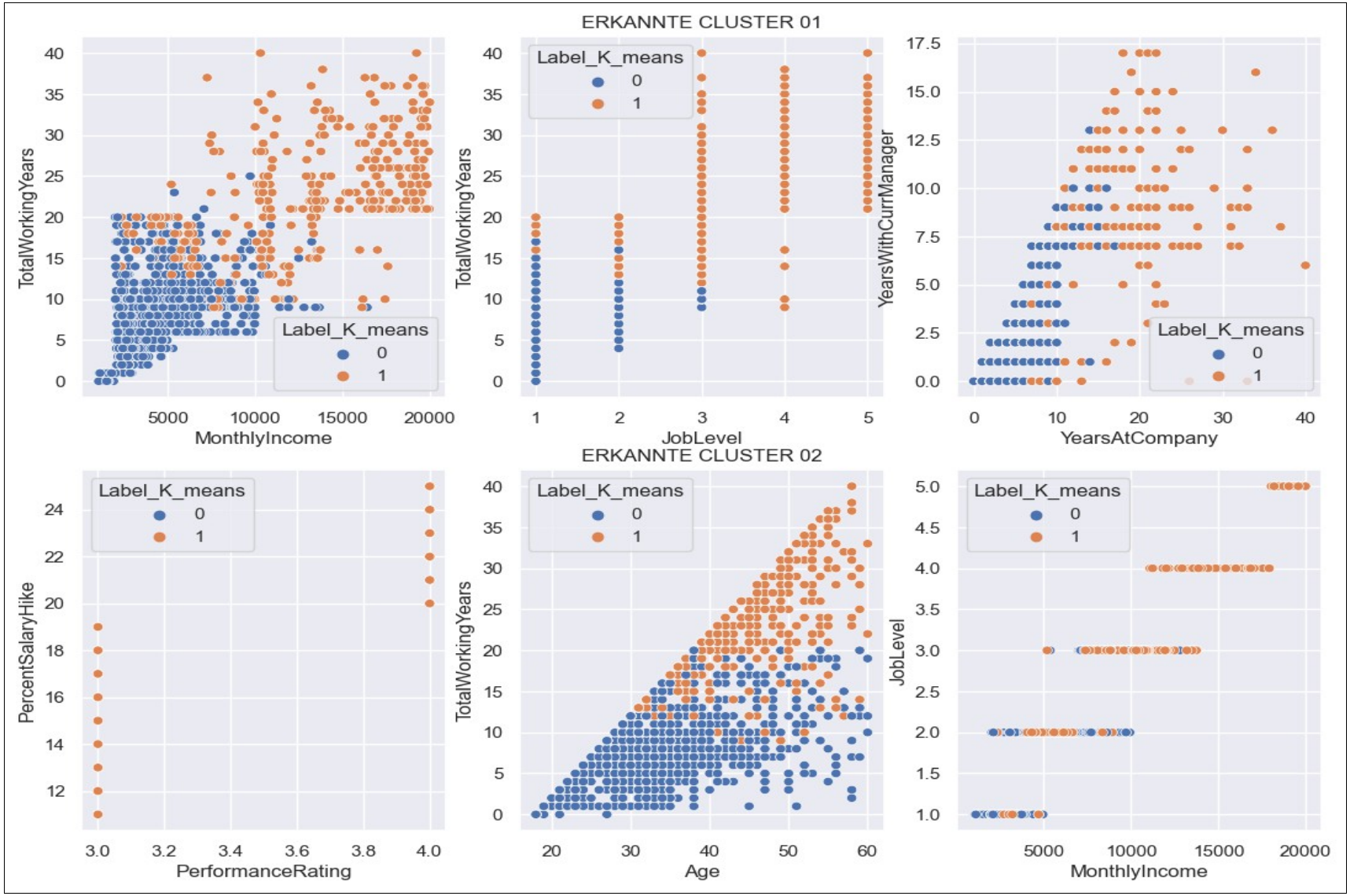
```
Mit 2 Cluster passend zu Attrition:

km = KMeans(n_clusters=2, random_state=33)
```

```
Trainieren und Vorhersage:

pred_km = km.fit_predict(x_pca)
```

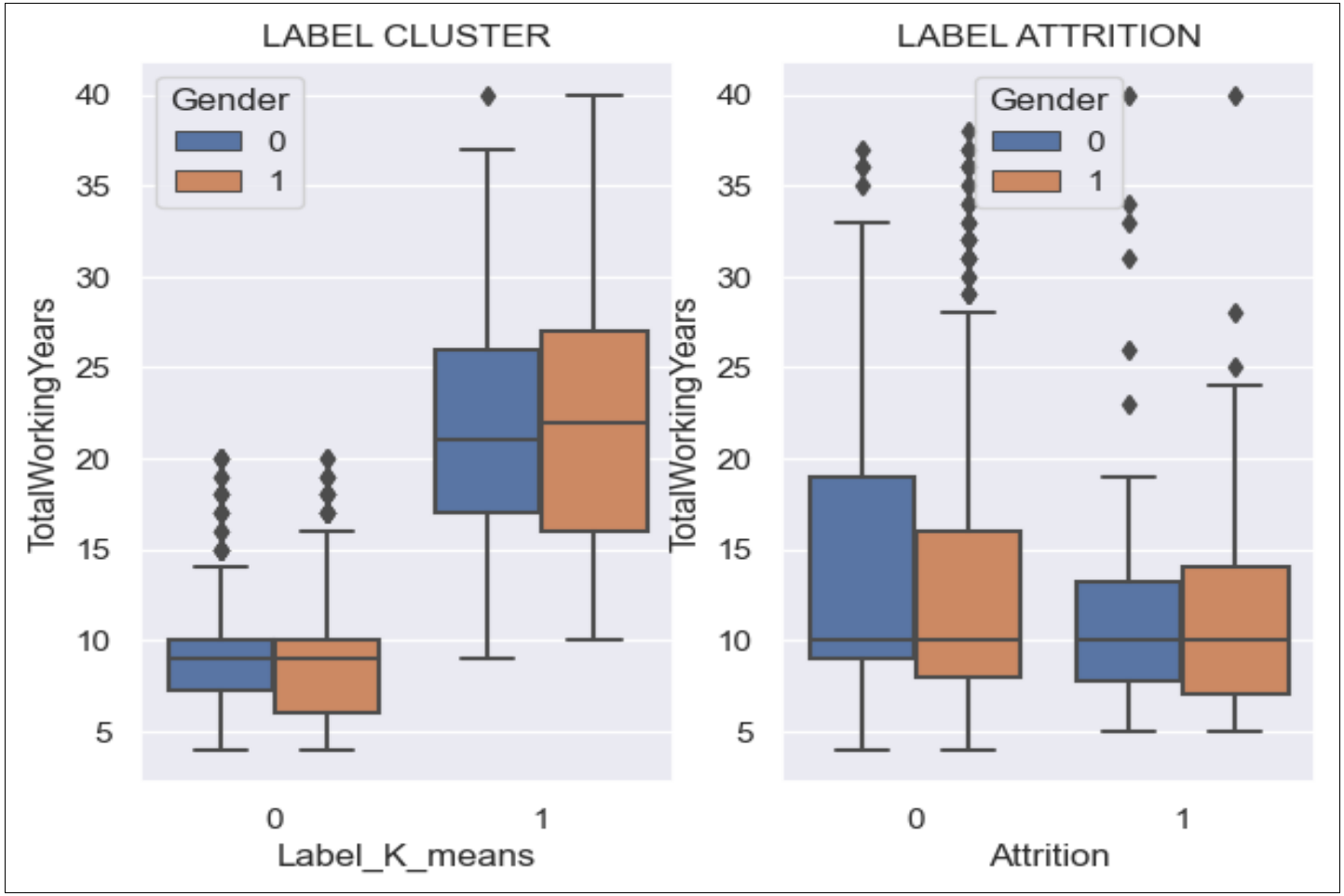
✗ neues Label mit den vorhergesagten Clustern an ursprünglichen Datensatz hinzufügen:

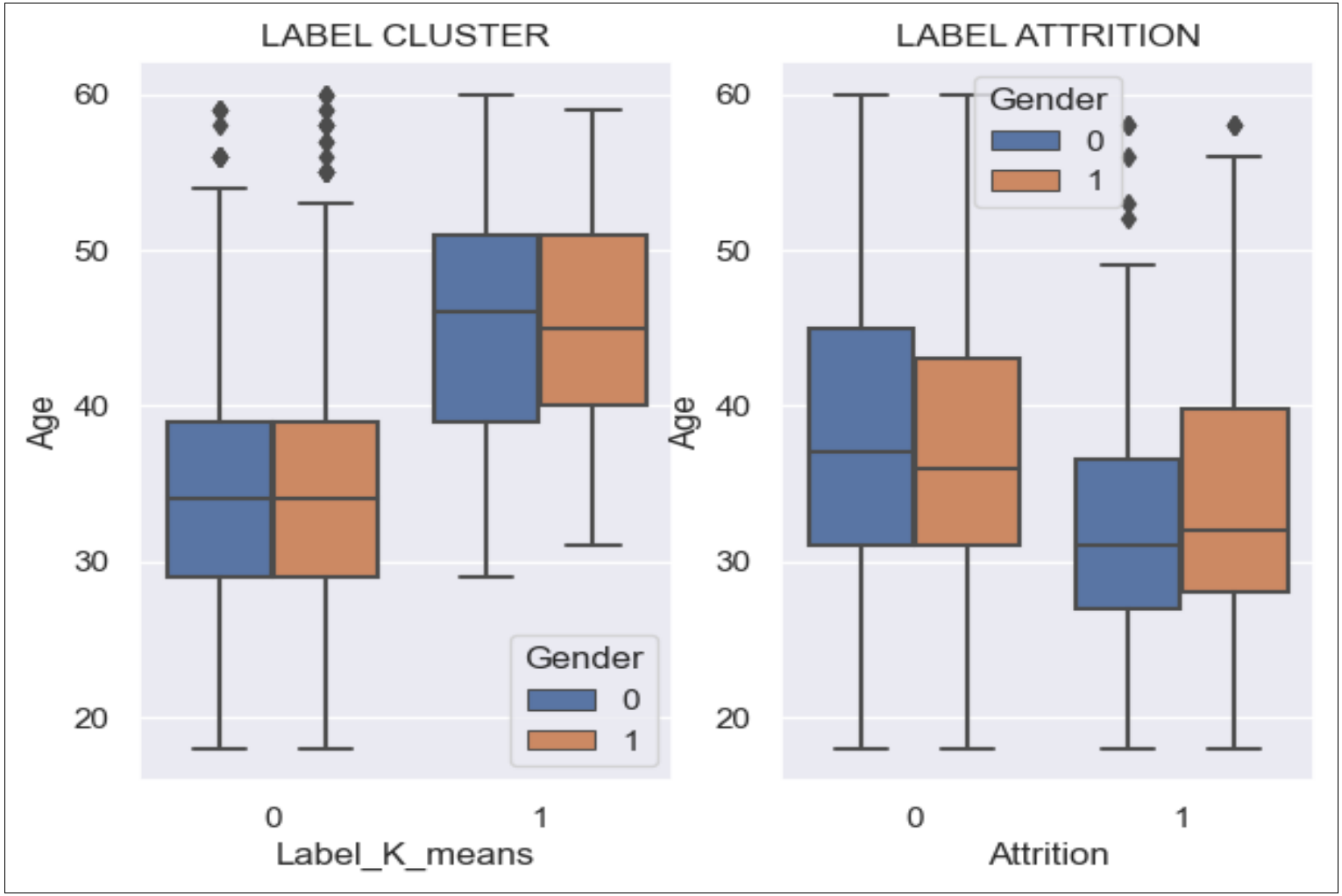
```
mitarbeiter02["Label_K_means"] = pred_km
```

✗ Visualisierung Cluster mit verschiedenen features in einem scatterplot:

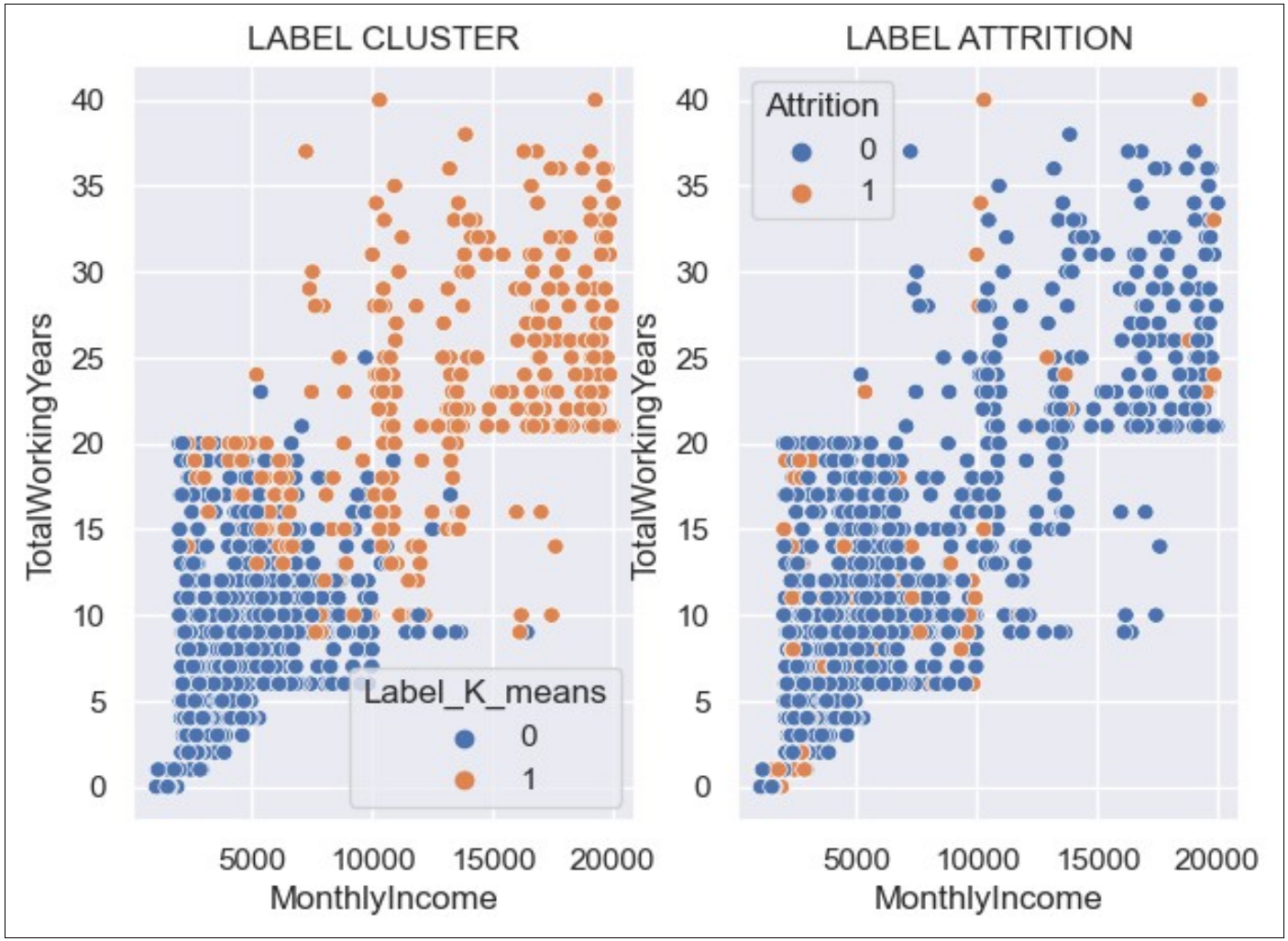


✗ Visualisierung Cluster mit verschiedenen features in boxplots:




LABEL CLUSTER
60
50
40
30
20
Age
Gender
0
1
0
1
Label_K_means
LABEL ATTRITION
60
50
40
30
20
Age
Gender
0
1
0
1
Attrition

✗ Vergleich ursprüngliches Label "Attrition" mit K-Means Label "Cluster":



✗ Vergleich Label "Attrition" mit Label "Cluster" mit PCA – classification report:

```
print(classification_report(mitarbeiter02["Attrition"], pred_km))
```

	precision	recall	f1-score	support
0	0.82	0.76	0.79	1233
1	0.08	0.11	0.10	237
accuracy			0.65	1470
macro avg	0.45	0.44	0.44	1470
weighted avg	0.70	0.65	0.67	1470

✗ K-MEANS OHNE PCA:

✗ K-Means mit allen features – Variable X01:

```
K-Means mit allen features und Datensatz X01 - alle features

km02 = KMeans(n_clusters=2, random_state=33)
pred_km02 = km02.fit_predict(X01)

print(classification_report(mitarbeiter02["Attrition"], pred_km02))
```

```
              precision    recall  f1-score   support

           0       0.82      0.76      0.79      1233
           1       0.10      0.14      0.12       237

    accuracy                           0.66      1470
   macro avg       0.46      0.45      0.45      1470
weighted avg       0.70      0.66      0.68      1470
```

✗ K-Means mit features mit den höchsten Korrelationen – Variable X02:

```
1  K-Means mit Datenssatz X02 - features mit höchsten Korrelationen
2
3  km03 = KMeans(n_clusters=2, random_state=33)
4  pred_km03 = km03.fit_predict(X02)
5
6  print(classification_report(mitarbeiter02["Attrition"], pred_km03))
```

```
              precision    recall  f1-score   support

           0       0.82      0.76      0.79      1233
           1       0.10      0.14      0.12       237

    accuracy                           0.66      1470
   macro avg       0.46      0.45      0.45      1470
weighted avg       0.70      0.66      0.68      1470
```

✗ K-Means mit features mit den geringsten Korrelationen – Variable X03:

```
K-Means mit Datenssatz X03 - features geringste Korrelationen
--> günstig wenn features keine hohen Korrelationen untereinander haben - bestes Ergebnis!!

km04 = KMeans(n_clusters=2, random_state=33)
pred_km04 = km04.fit_predict(X03)

print(classification_report(mitarbeiter02["Attrition"], pred_km04))

```

```
              precision    recall  f1-score   support

           0       0.85      0.76      0.80      1233
           1       0.20      0.32      0.25       237

    accuracy                           0.69      1470
   macro avg       0.53      0.54      0.53      1470
weighted avg       0.75      0.69      0.71      1470
```

FAZIT:

- der Datensatz war sehr gut zu bearbeiten – überschaubare Anzahl an features, keine Nullwerte und ein vorgegebenes label

- Ergebnisse Supervised Learning:

  - beste Ergebnisse unter Einbezug aller features – X01 oder unter Einbezug der features mit den höchsten Korrelationen – X02

  - beste Ergebnisse in Bezug auf die Algorithmen:

    SVC und Logistische Regression

  - auch die Ergebnisse mit den anderen Algorithmen(KNN, NaiveBayes, RandomForest) sind sehr gut

  - Ergebnisse unter Einbezug der features mit den geringsten Korrelationen – X03 – mittelmäßige/gute Ergebnisse

➢ Ergebnisse Unsupervised Learning:

- mit PCA reduzierte features – 1 Dimension:

  Ergebnisse vergleichbar mit X03 (features mit geringsten Korrelationen) – mittelmäßige/gute Ergebnisse

- K-Means:

- K-Means lieferte meiner Meinung nach keine guten Ergebnisse und ist für die **Fragestellung „Kündigung Ja/Nein“ nicht geeignet**

  Gründe:

  - ✔ die Stärke des K-Means liegt in einer Art Segmentierung von Datenbereichen wie z.B. bei Bildern, Kundengruppen, Produktgruppen usw. Das Label „Attrition“ in diesem Datensatz hingegen beinhaltet eine Klassifizierung **„Kündigung Ja/Nein“**, deshalb eher ungeeignet für den K-Means

- ✔ Ausreißer oder Anomalien spielen in diesem Datensatz keine große Rolle, für die der K-Means normalerweise effektiv eingesetzt werden kann

- ✔ günstig für den K-Means sind außerdem Daten, die untereinander kaum oder nur schwach korrelieren, dadurch erhält man bessere Ergebnisse

- Der Durchlauf mit allen Trainings- und Testvariablen von X01 – X03 als auch mit der PCA reduzierten Variablen X04 lieferten keine guten Ergebnisse – kein klares Clustering erkennbar

- Am besten schnitt noch die Trainings- und Testvariable X03 ab, in der die features mit den niedrigsten Korrelationen enthalten waren, was meiner Recherche nach auch günstig ist für die Anwendung des K-Means

Vergleich Algorithmen Supervised Learning

```
ALLE FEATURES:

Genauigkeit LogReg01:        91.16%
Genauigkeit KNN01    :       90.48%
Genauigkeit SVC01    :       91.50%
Genauigkeit NAIVE01  :       84.01%
Genauigkeit Random01:        89.12%

FEATURES MIT HÖCHSTEN KORRELATIONEN:

Genauigkeit LogReg02:        91.50%
Genauigkeit KNN02    :       89.46%
Genauigkeit SVC02    :       90.14%
Genauigkeit NAIVE02  :       82.31%
Genauigkeit Random02:        89.12%

FEATURES MIT GERINGSTEN KORRELATIONEN:

Genauigkeit LogReg03:        88.78%
Genauigkeit KNN03    :       88.10%
Genauigkeit SVC03    :       88.78%
Genauigkeit NAIVE03  :       88.78%
Genauigkeit Random03:        88.78%

FEATURES MIT PCA REDUZIERT:

Genauigkeit LogReg04:        88.78%
Genauigkeit KNN04    :       88.10%
Genauigkeit SVC04    :       88.78%
Genauigkeit NAIVE04  :       88.78%
Genauigkeit Random04:        73.81%
```

Vergleich Label „Attrition“ und neues K-Means-Label PCA-reduziert

```
print(classification_report(mitarbeiter02["Attrition"], pred_km))
```

```
              precision    recall  f1-score   support

           0       0.82      0.76      0.79      1233
           1       0.08      0.11      0.10       237

    accuracy                           0.65      1470
   macro avg       0.45      0.44      0.44      1470
weighted avg       0.70      0.65      0.67      1470
```

Vergleich Label „Attrition“ und neues K-Means-Label x03 – bestes Ergebnis

```
K-Means mit Datenssatz X03 - features geringste Korrelationen
--> günstig wenn features keine hohen Korrelationen untereinander haben - bestes Ergebnis!!

km04 = KMeans(n_clusters=2, random_state=33)
pred_km04 = km04.fit_predict(X03)

print(classification_report(mitarbeiter02["Attrition"], pred_km04))

```

```
              precision    recall  f1-score   support

           0       0.85      0.76      0.80      1233
           1       0.20      0.32      0.25       237

    accuracy                           0.69      1470
   macro avg       0.53      0.54      0.53      1470
weighted avg       0.75      0.69      0.71      1470
```