Mitarbeiterabwanderung

Machine Learning Projekt: Ein binäres Klassifikationsproblem

vorgelegt am 29. Juli 2022

Fakultät Wirtschaft

Studiengang Wirtschaftsinformatik

Kurs WWI2019F

von

Laura Morlok

Inhaltsverzeichnis

1. Zielsetzung 4
2. Datensatzbeschreibung und Qualitätssicherung 4
3. Pre-Processing 5
   3.1 Feature Selection 5
   3.2 Label Encoding und One Hot Encoding sowie Normalisierung der Skalen 6
4. Entwicklung des Modells 7
   4.1 K-Nearest-Neigbor (KNN) 7
   4.2 Random Forest 11
5. Diskussion des Ergebnisses 14

Abbildungsverzeichnis

Abb. 1: Datenvisualisierung _____ 5
Abb. 2: Ergebnis des Chi-Square Tests _____ 6
Abb. 3: Accuracy nach Anzahl der Nachbarn (KNN) _____ 8
Abb. 4: Konfusionsmtarix und Klassifikationsreport für k=4 (KNN) _____ 8
Abb. 5: F1-Score nach Anzahl der Nachbarn (KNN) _____ 9
Abb. 6: Konfusionsmatrix für k=3 (KNN) _____ 9
Abb. 7: Matthews Correlation Coefficient nach Anzahl der Nachbarn (KNN) _____ 10
Abb. 8: Resampled Dataset: Accuracy nach Anzahl der Nachbarn (KNN) _____ 10
Abb. 9: Resampled Dataset: Konfusionsmatrix und Klassifikationsreport für k=2 (KNN) _____ 11
Abb. 10: Resampled Dataset: Konfusionsmatrix und Klassifikationsreport für Standardhyperparameter (RF) _ 12
Abb. 11: Ergebnis des random_grid (RF) _____ 12
Abb. 12: Erstellung des param_grid (RF) _____ 13
Abb. 13: Ergebnis mit dem param_grid (RF) _____ 13
Abb. 14: Feature Importance (RF) _____ 13

1. Zielsetzung

Die **Mitarbeiterbindung** ist für viele Unternehmen im Zeitalter der Digitalisierung, der Globalisierung und des schnellen Wandels ein kritischer Erfolgsfaktor. Es gibt häufig eine Vielzahl von Gründen, weshalb Mitarbeitende ihr Unternehmen verlassen.
Die Frage, die sich hier stellt, ist:
Welche Faktoren tragen besonders dazu bei, dass Mitarbeitende ihr Unternehmen verlassen?
Die Beantwortung dieser Frage hilft den Personalbereichen der jeweiligen Unternehmen, ihre Strategien zur Mitarbeiterbindung konkreter zu definieren und so die Unternehmensaustritte proaktiv zu reduzieren.
Um die Personalplanung zu unterstützen, liegt das Ziel der Arbeit darin, die Schlüsselfaktoren des Unternehmensaustritts identifizieren zu können. Es wird ein Modell trainiert, welches klassifizieren kann, welcher Mitarbeitende in dem Unternehmen bleibt und welcher das Unternehmen verlässt. Dies hilft Unternehmen bei ihrer Planungssicherheit und ermöglicht es, konkret auf Mitarbeitende zuzugehen und abgestimmte Maßnahmen einzuleiten, um einen Unternehmensaustritt zu verhindern. Die ermittelte Gesamtzahl der Mitarbeitenden, die das Unternehmen verlassen, hilft der Personalplanung, mit einer gewissen Zahl an Mitarbeitenden zu rechnen, die das Unternehmen verlassen werden.

2. Datensatzbeschreibung und Qualitätssicherung

Der Datensatz **„IBM HR Analytics Employee Attrition & Performance“** entstammt der Plattform Kaggle. Kaggle ist eine Online-Community, die der Google LLC angehört und primär der Organisation von Data-Science Wettbewerben dient. Die Plattform enthält eine Vielzahl an veröffentlichten Datensätzen.
Der für dieses Projekt gewählte Datensatz beinhaltet fiktionale Daten über Arbeitnehmende eines Unternehmens. IBM Data Scientists erstellten diesen Datensatz, da echte Mitarbeiterdaten aus datenschutzrechtlichen Gründen in dieser Form oftmals nicht veröffentlicht werden können. Der Datensatz beinhaltet 35 Spalten, die persönliche sowie berufsbezogene Daten enthalten, sowie 1470 Zeilen.
Man kann zwischen kategorialen sowie numerischen Variablen unterscheiden. Es sind keine NaN-Werte im Datensatz enthalten.

Eine erste Analyse der numerischen Daten zeigt, dass die durchschnittliche Person 37 Jahre alt ist. Sie ist bereits seit elf Jahren arbeitstätig, davon sieben Jahre in dem gegenwärtigen Unternehmen. Die jüngste Person ist 18 Jahre alt, die älteste Person 60. Im Schnitt wohnen die Mitarbeitenden rund neun Kilometer von der Arbeitsstätte entfernt.

Anhand der statistischen Auswertung der numerischen Daten konnten insgesamt vier Spalten identifiziert werden, die **keine Bedeutung** für die weitere Arbeit haben. Diese sind im Folgenden aufgelistet und werden aus dem Datensatz **entfernt**:

- **Over18:** Da anhand der statistischen Auswertung die jüngste Person als 18 identifiziert wurde.
- **StandardHours:** Da anhand der statistischen Auswertung erkannt wurde, dass diese ausnahmslos 80 Stunden betragen.

- **EmployeeCount:** Da anhand der statistischen Auswertung erkannt wurde, dass diese ausnahmslos eins beträgt.
- **EmployeeNumber**

Die **Erstellung von Diagrammen** hilft, um weitere Erkenntnisse über die Daten zu gewinnen.

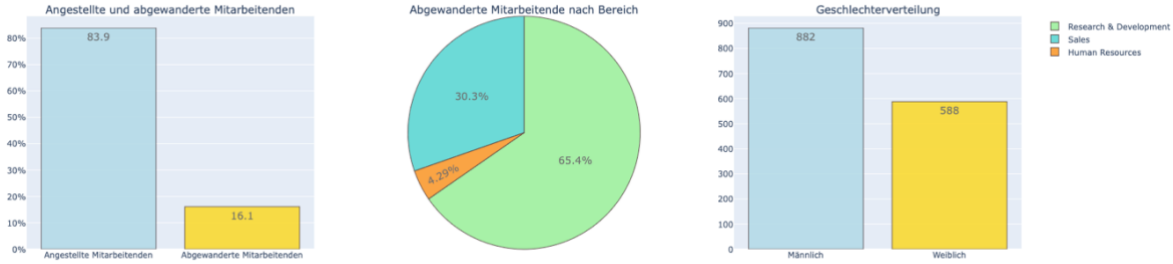

Abb. 1: Datenvisualisierung

Das **erste Balkendiagramm** zeigt das Verhältnis von angestellten und abgewanderten Mitarbeitenden des Unternehmens auf. 83,9 Prozent der Personen sind gegenwärtig in dem Unternehmen angestellt, 16,1 Prozent haben das Unternehmen bereits verlassen.

Das **zweite Kuchendiagramm** bildet die Bereiche ab, in denen die abgewanderten Mitarbeitenden tätig waren. 65,4 Prozent von ihnen waren in dem Bereich Research&Development tätig, 30,3 Prozent im Vertrieb sowie 4,2 Prozent im Personalwesen. Da mit 961 Personen allerdings auch wesentlich mehr Mitarbeitende im Research&Development Bereich im Gegensatz zu den anderen Bereichen (Vertrieb = 446 und Personalwesen = 63) tätig sind, dienen diese Informationen rein deskriptiven Zwecken.

Das **dritte Balkendiagramm** zeigt die Geschlechterverteilung auf. Auffällig ist, dass Männer mit 882 Einträgen wesentlich stärker vertreten sind als Frauen mit 588 Einträgen.

3. Pre-Processing

3.1 Feature Selection

Die Qualität der Inputvariablen hat einen maßgeblichen Einfluss auf die Leistung und die Qualität der Machine Learning Modelle.
Um die **Vorhersagequalität möglichst zu maximieren,** werden daher die **kategorialen Variablen auf ihre Relevanz für die Zielerreichung geprüft.** Die Variablen ohne Einfluss auf die Zielvariable werden aus dem Datensatz entfernt. Durch die Reduzierung der Features auf die relevantesten werden die Anforderungen an Daten und Ressourcen (Curse of Dimensionality) reduziert sowie die Interpretierbarkeit des Modells erhöht. Dieser Schritt ist demnach ein sehr wichtiger Teil des Machine Learning Prozesses.

Um nun herauszufinden, welche der insgesamt 15 kategorialen Variablen die Abwanderung beeinflussen, werden nachfolgend **ein Chi-Square-Test** durchgeführt.

Die Voraussetzungen für den Test bestehen darin, dass die Variablen nominal oder ordinal skaliert sind, die Stichprobe zufällig gezogen wurde sowie größer 50 ist. Diese Voraussetzungen werden durch den gewählten Datensatz erfüllt.
Chi-Square-Tests werden angewandt, um herauszufinden, welche der Variablen Einfluss auf die Zielvariable haben. Hierbei findet der **Unabhängigkeitstest** Anwendung. Die abhängige Zielvariable stellt die Spalte **Attrition** dar. Geprüft wird nun der Einfluss der unabhängigen Variablen wird auf diese abhängige Variable.

Die Nullhypothese H_0 besteht in der Annahme, dass die unabhängige Variable keinen Einfluss auf die abhängige Variable, also auf die Abwanderung, aufweist. Die Gegenhypothese H_1 besteht darin, dass die unabhängige Variable die Abwanderung beeinflusst.

H_0: Die unabhängige Variable hat keinen Einfluss auf die abhängige Variable.
H_1: Die unabhängige Variable hat einen Einfluss auf die abhängige Variable.

Ist der p-Wert > 0,5, so wird die Nullhypothese bestätigt. Ist p < 0,5, so wird die Gegenhypothese H_1 bestätigt.

Folgende Variablen werden auf ihren Einfluss und damit auf ihre Relevanz geprüft:
BusinessTravel, Department, Education, EducationField, EnvironmentSatisfaction, Gender, JobInvolvement, JobLevel, JobRole, JobSatisfaction, MaritalStatus, OverTime, PerformanceRating, RelationshipSatisfaction, WorkLifeBalance.

Der folgende Auszug zeigt alle Variablen mit p > 0,5. Für diese Variablen wird H_0 bestätigt.

```
Das Attribut Education hat keinen Einfluss auf die Abwanderung. Der p-Wert beträgt 0.5455.
Das Attribut Gender hat keinen Einfluss auf die Abwanderung. Der p-Wert beträgt 0.2588.
Das Attribut PerformanceRating hat keinen Einfluss auf die Abwanderung. Der p-Wert beträgt 0.9118.
Das Attribut RelationshipSatisfaction hat keinen Einfluss auf die Abwanderung. Der p-Wert beträgt 0.1550.
```

Abb. 2: Ergebnis des Chi-Square Tests

Für **Education, Gender, PerformanceRating und RelationshipSatisfaction** wird die Nullhypothese aufgrund eines p-Wertes > 0,5 bestätigt, weshalb diese vier Variablen aus dem Datensatz entfernt werden.

3.2 Label Encoding und One Hot Encoding sowie Normalisierung der Skalen

Um ein Klassifikationsmodell trainieren zu können, müssen nun auch die **kategorialen Variablen in numerische Form** gebracht werden. Dies geschieht, indem die Ausprägungen der Variablen mit 0 und 1 gekennzeichnet werden. 0 steht für nichtzutreffend und 1 für zutreffend. Für die binäre Variable Attrition wird dies mit **Label Encoding** erreicht, für die restlichen Variablen, die alle mehr als zwei Ausprägungen aufweisen, wird dies mithilfe von **One Hot Encoding** durchgeführt**.**

Da die Skalen der Attribute in dem Datensatz sehr unterschiedlich sind (z.B Age: 18-60 und MonthlyIncome: 1.009 – 19.999), werden diese mithilfe des MinMaxScalers auf einer Skala von null bis eins normalisiert. Da einige Algorithmen auf den Abständen zwischen den Datenpunkten aufbauen, ist die Normalisierung der Skalen bei geplanter Nutzung dieser Algorithmen ein wichtiger Pre-Processing Schritt. Die Normalisierung betrifft lediglich die

numerischen Attribute, die keine binäre Ausprägung aufweisen. Aus diesem Grund entfallen die Label Encoded und die One Hot Encoded Attribute.

4. Entwicklung des Modells

Aus der Zielsetzung heraus findet eine **binäre Klassifikation** Anwendung. Die Klassifikation gehört neben der Regression zum überwachten Lernen und lässt sich als Zuordnung von Beobachtungen in vordefinierte Klassen beschreiben. Im vorliegenden Fall bilden *Unternehmensaustritt* sowie *kein Unternehmensaustritt* die Klassen 0 und 1.

4.1 K-Nearest-Neigbor (KNN)

Der KNN Algorithmus ist ein **überwachtes Lernverfahren,** der das Konzept der Nähe nutzt, um Klassifizierungen oder Vorhersagen über die Gruppierung eines einzelnen Datenpunktes zu treffen. Er kann sowohl für Regressions- als auch für Klassifikationsprobleme verwendet werden, wird aber primär für Klassifizierungen genutzt.
Bei Klassifikationsproblemen wird eine Bezeichnung für eine Klasse auf der Grundlage einer **Mehrheitswahl** zugewiesen, d. h. es wird die Bezeichnung verwendet, die um einen bestimmten Datenpunkt herum am häufigsten vorkommt.
Um zu ermitteln, welche Datenpunkte einem bestimmten Abfragepunkt am nächsten liegen, muss der Abstand zwischen dem Abfragepunkt und den anderen Datenpunkten berechnet werden.
Der k-Wert im KNN-Algorithmus legt fest, wie viele Nachbarn geprüft werden, um die Klassifizierung eines bestimmten Abfragepunkts zu bestimmen.
Insgesamt ist es empfehlenswert, eine ungerade Zahl für k zu wählen, um „Unentschieden" in der Klassifizierung zu vermeiden.
Die Wahl für den ersten Algorithmus fällt auf den KNN Algorithmus, da dieser von der Funktionsweise leicht nachvollziehbar und leicht implementierbar ist. Er besitzt wenige Hyperparameter und liefert oftmals auch bei vielen Inputparametern gute Ergebnisse.

Im ersten Schritt wird der Datensatz in ein **Trainings- und ein Testdatensatz** geteilt. Der Trainingsdatensatz entspricht hierbei 70 Prozent des ursprünglichen Datensatzes und enthält somit 1029 Zeilen, der Testdatensatz entspricht 30 Prozent des ursprünglichen Datensatzes und enthält somit 441 Zeilen.
Der Parameter *stratify* sorgt dafür, dass Klasse 0 und Klasse 1 jeweils zu 70% im Trainingsdatensatz und zu 30% im Testdatensatz vorkommen.

4.1.1 Metrik: Accuracy

Im nächsten Schritt wird das Modell trainiert. Es wird die Nachbaranzahl *k* gesucht, für welche die Metrik Accuracy am höchsten ist.

Die **Accuracy** stellt eine beliebte Metrik dar, die leicht verständlich ist und daher einen guten ersten Anhaltspunkt für die Performance eines Modells bietet. Sie gibt Auskunft über die **Trefferwahrscheinlichkeit** und wird wie folgt berechnet:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Das folgende Schaubild zeigt die Abhängigkeit der Accuracy von der Anzahl der Nachbarn:

```
Die Anzahl der Nachbarn mit der höchsten Accuracy mit 0.8481 ist: 4.
```

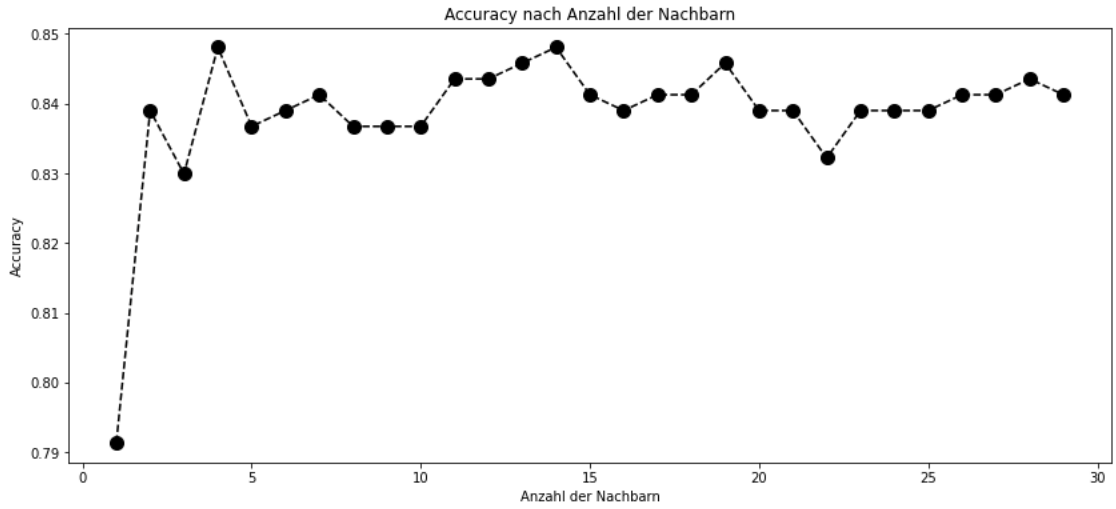

Abb. 3: Accuracy nach Anzahl der Nachbarn (KNN)

Wie auf dem Diagramm erkennbar ist, kann die höchste Accuracy mit **0.8481** mit **vier Nachbarn** erreicht werden. Damit ist dieses Modell lediglich gering besser als ein Modell, welches vorhersagt, dass alle Mitarbeitenden in dem Unternehmen bleiben. Aus diesem Grund wird das Modell mit vier Nachbarn trainiert mithilfe der Konfusionsmatrix sowie dem Klassifikationsreport näher analysiert. Diese sind auf der nachfolgenden Abbildung dargestellt.

```
[[365   5]
 [ 62   9]]

True Positives stehen für die Mitarbeitenden, die das Unternehmen nicht verlassen.

True Positive(TP)  =  365
False Negative(FN) =  5
False Positive(FP) =  62
True Negative(TN)  =  9

              precision    recall  f1-score   support

           0       0.85      0.99      0.92       370
           1       0.64      0.13      0.21        71

    accuracy                           0.85       441
   macro avg       0.75      0.56      0.56       441
weighted avg       0.82      0.85      0.80       441
```

Abb. 4: Konfusionsmtarix und Klassifikationsreport für k=4 (KNN)

Anhand der Konfusionsmatrix wird erkennbar, dass es insgesamt 62 False Positives sowie fünf False Negatives gibt. Da der Datensatz mit 370 Einträgen der Klasse 0 und 71 Einträgen der Klasse 1 sehr unausgeglichen ist und *k*=4 entspricht, wird die Mehrheit der zu

klassifizierenden Datenpunkte der Klasse 0 zugeordnet. Dies erklärt auch, weshalb der F1-Score zwar für die Klasse 0 mit 0.92 sehr hoch ist, für die Klasse 1 mit 0.21 jedoch äußerst gering. Dies ist ein Beispiel des **Overfittings.**
Es lässt sich festhalten, dass die Accuracy mit 0.8481 ohne Vorkenntnisse über den Datensatz als recht gut erscheint. Doch bei einer genaueren Analyse wird deutlich, dass die Accuracy keine geeignete Metrik für unausgeglichene Datensätze darstellt. Aus diesem Grund wird im Folgenden der F1-Score näher betrachtet.

4.1.2 Metrik: F1-Score

Der beste F1-Score mit 0.8078 wird bei *k*=3 erreicht. Dies wird auf dem nachfolgenden Diagramm dargestellt.

```
Die Anzahl der Nachbarn mit dem höchsten F1-Wert mit 0.8078 ist: 3.
```

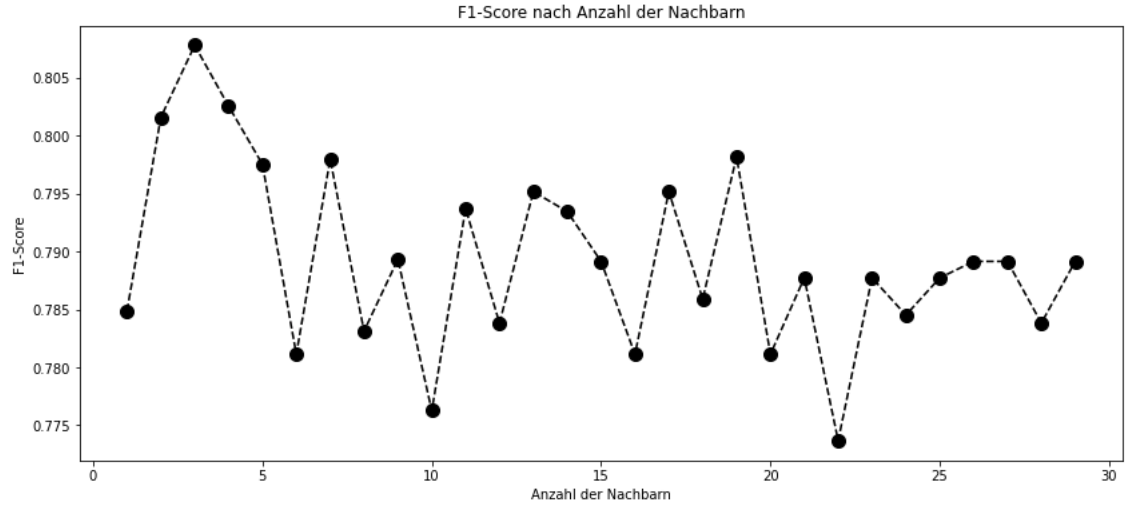

Abb. 5: F1-Score nach Anzahl der Nachbarn (KNN)

Die zugehörige Konfusionsmatrix sieht wie folgt aus:

```
[[349  21]
 [ 54  17]]
```

Abb. 6: Konfusionsmatrix für k=3 (KNN)

Da der Matthews Korrelationskoeffizient für unausgeglichene Datensätze besser geeignet ist als die bisher betrachteten Metriken, wird nun die Anzahl der Nachbarn gesucht, für die der MCC am höchsten ist. Das folgende Diagramm zeigt das Ergebnis:

```
Die Anzahl der Nachbarn mit dem höchsten MCC-Wert mit 0.2393 ist: 3.
```

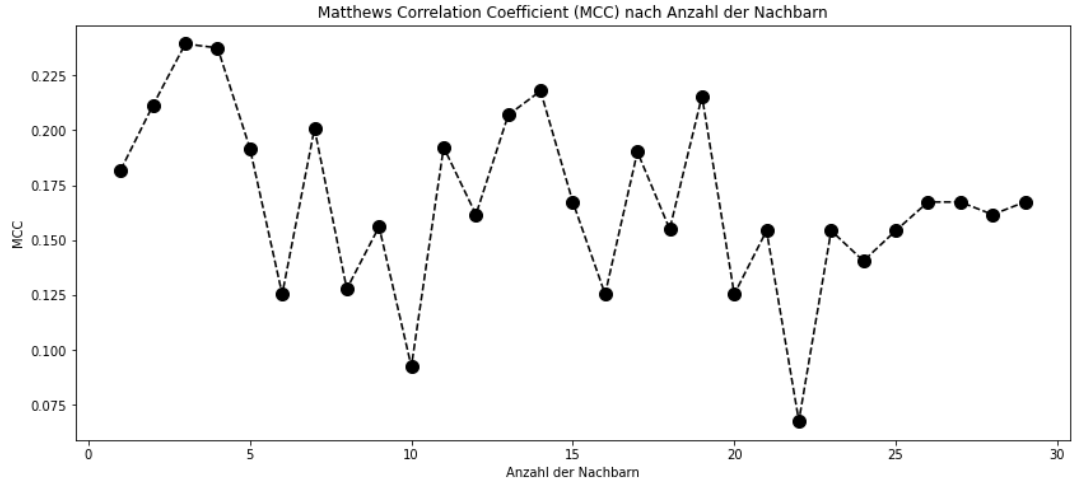

Abb. 7: Matthews Correlation Coefficient nach Anzahl der Nachbarn (KNN)

Mit einem Ergebnis von 0,2393 bei drei Nachbarn deutet das Ergebnis nur auf einen sehr leichten Zusammenhang von vorhergesagten und tatsächlichen Werten hin.

Um bessere Ergebnisse mit dem KNN-Algorithmus zu erhalten, muss die **Gleichverteilung der Klassen** in dem Datensatz verbessert werden.
Um den unausgeglichenen Daten entgegenzuwirken, kann entweder Under- oder Oversampling Anwendung finden. Da der Trainingsdatensatz mit insgesamt 1029 Einträgen nicht sonderlich groß ist, und die Undersampling Methode die Mehrheitsklasse auf die Größe der Minderheitsklasse reduzieren würde, wird Oversampling angewandt. Hierfür wird die **Synthetic Minority Oversampling Technique (SMOTE)** mit dem Parameters *minority* angewandt, sodass die Minderheit aufgefüllt wird. Nach Anwendung der SMOTE enthält der Trainingsdatensatz sowohl 863 Objekte der Klasse 0 sowie 863 Objekte der Klasse 1.

Da nun ein ausgeglichener Datensatz vorliegt, wird erneut die höchste Accuracy gesucht.

```
Resampled dataset shape: Counter({0: 863, 1: 863})
Die Anzahl der Nachbarn mit der höchsten Accuracy mit 0.7551 ist: 2.
```

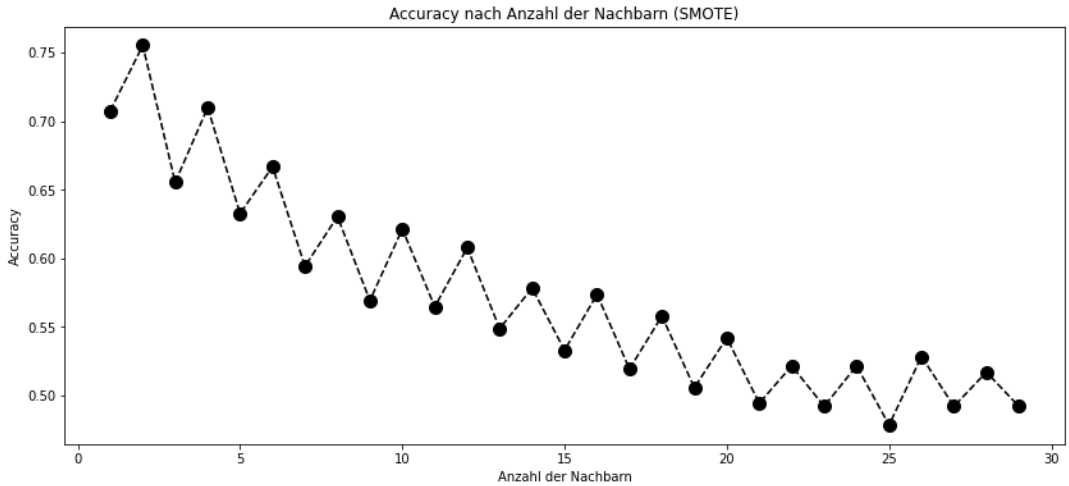

Abb. 8: Resampled Dataset: Accuracy nach Anzahl der Nachbarn (KNN)

```
[[155 215]
 [  9  62]]

True Positive(TP)  = 155
False Negative(FN) =  215
False Positive(FP) =  9
True Negative(TN)  =  62

              precision    recall  f1-score   support

           0       0.95      0.42      0.58       370
           1       0.22      0.87      0.36        71

    accuracy                           0.49       441
   macro avg       0.58      0.65      0.47       441
weighted avg       0.83      0.49      0.54       441


Matthews Korrelationskoeffizient: 0.2222.
```

Abb. 9: Resampled Dataset: Konfusionsmatrix und Klassifikationsreport für k=2 (KNN)

Die höchste Accuracy liegt mit 0,7551 bei *k*=2 und ist somit niedriger als zuvor. Mit insgesamt 215 False Negatives und 9 True Positives besteht nun nicht mehr das Problem, dass das Modell die Mehrheit der Objekte der Klasse 0 zuordnet, dennoch werden 224 von 441 Objekte falsch klassifiziert. Der Matthews Korrelationskoeffizient beträgt 0,2222.
Die Zackenform des Diagrammes lässt zudem erahnen, dass die Accuracy für eine ungerade Anzahl an Nachbarn aufgrund des Mehrheitsentscheids geringer ist als die Accuracy bei einer ungeraden Anzahl an Nachbarn.

Es lässt sich festhalten, dass mit der Gleichverteilung der Klassen in dem Trainingsdatensatz die Performance des Modells sinkt. Aus diesem Grund wird nun ein zweiter Algorithmus, der **Random Forest Algorithmus**, angewandt. Für diesen Algorithmus findet im Folgenden ausschließlich der mit SMOTE erweiterte Trainingsdatensatz Anwendung, da sich der Random Forest Algorithmus nicht für unausgeglichene Datensätze eignet.

4.2 Random Forest

Der Random Forest Algorithmus zählt, wie der KNN Algorithmus, zu den überwachten Lernverfahren. Hierbei werden mehrere unkorrelierte, unabhängig entwickelte, zufällige Entscheidungsbäume erstellt. Jeder Baum trifft eine einzelne Entscheidung, aus der Mehrheit der Einzelentscheidungen wird dann die endgültige Entscheidung abgeleitet. Die Entscheidungen eines Random Forest sind dabei im Gegensatz zu neuronalen Netzen nachvollziehbar und leicht untersuchbar.
Ein großer Vorteil des Algorithmus liegt in dem schnellen Training, die Trainingszeit wächst linear mit der Anzahl der Bäume. Zudem erfolgt die Evaluation unabhängig voneinander und kann daher parallelisiert werden. Ein weiterer Vorteil liegt in der geringeren Anfälligkeit für Overfitting, da auf zufälligen Subsamples mit zufälligen Variablenauswahl trainiert wird. Da das Overfitting bei dem KNN-Algorithmus ein Problem darstellte und der Random Forest Algorithmus nicht auf Distanzen aufbaut, fällt die Wahl auf diesen Algorithmus.

4.2.1 Metrik: Accuracy

Zunächst wird ein Modell mit den standardmäßig eingetragenen Hyperparametern trainiert.

Hierbei kann eine Accuracy von 0.8776 erreicht werden.

```
Die Accuracy des Random Forest Algorithmus beträgt: 0.8776.

True Positive(TP)  =  362
False Negative(FN) =  8
False Positive(FP) =  46
True Negative(TN)  =  25

              precision    recall  f1-score   support

           0       0.89      0.98      0.93       370
           1       0.76      0.35      0.48        71

    accuracy                           0.88       441
   macro avg       0.82      0.67      0.71       441
weighted avg       0.87      0.88      0.86       441
```

Abb. 10: Resampled Dataset: Konfusionsmatrix und Klassifikationsreport für Standardhyperparameter (RF)

4.2.2 Hyperparameter Tuning

Es gibt eine Vielzahl an Hyperparametern, die bei dem Random Forest Algorithmus gesetzt werden können. Um nun die besten Hyperparameter für das Modell zu ermitteln, wird ein sogenannten *random_grid* erzeugt, welches für jeden Hyperparameter, die wir betrachten, mehrere Werte enthält. Wir betrachten folgende sechs Hyperparameter:

- n_estimators: Anzahl der Bäume
- max_feature: Maximale Anzahl an Features, die bei einem Split betrachtet werden
- max_depth: Maximale Anzahl an Ebenen
- min_samples_split: Mindestanzahl an Einträgen in einem Blatt, bevor dieses geteilt wird
- min_samples_leaf: Mindestanzahl an Einträgen in einem Blatt
- bootstrap: Sampling-Methode

Nun wird mithilfe des *random_grid* iterativ nach dem Modell mit dem besten Ergebnis und den dazugehörigen Hyperparametern gesucht. Das Ergebnis ist wie folgt:

```
Fitting 3 folds for each of 50 candidates, totalling 150 fits
0.9328381642512077
{'n_estimators': 400, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'max_depth': 10, 'bootstrap': True}
```

Abb. 11: Ergebnis des random_grid (RF)

Nun wird ein *param_grid* erstellt, welches neue Werte rund um die gefundenen Werte enthält. Darauf wird erneut trainiert, um so das beste Ergebnis mit den dazugehörigen Hyperparametern zu finden. Das param_grid sieht wie folgt aus:

```
param_grid = {
    'bootstrap': [True],
    'max_features': ['sqrt'],
    'max_depth': [5, 10, 15],
    'min_samples_leaf': [1, 2 , 3],
    'min_samples_split': [2 ,3, 4],
    'n_estimators': [200, 400, 600, 800]
}
# Create a based model
rf = RandomForestClassifier()

# Instantiate the grid search model
grid_search = GridSearchCV(estimator = rf, param_grid = param_grid,
                           cv = 2, n_jobs = -1, verbose = 2)
```

Abb. 12: Erstellung des param_grid (RF)

```
Fitting 2 folds for each of 108 candidates, totalling 216 fits
0.9328381642512077
```

Abb. 13: Ergebnis mit dem param_grid (RF)

Das Ergebnis nach zwei Schritten des Hyperparametertunings liegt bei **93,28 Prozent Accuracy**. Größere Skalen in dem param_grid sowie mehr Blöcke pro Kreuzvalidierung (folds) könnten das Ergebnis noch verbessern. Aufgrund von dem Ressourcenanspruch wird das vorliegende Ergebnis als **endgültiges Ergebnis** akzeptiert.

Der Random Forest Algorithmus bietet eine Möglichkeit, die Wichtigkeit der Inputvariablen auf die Zielvariable zu messen. Hierfür wird erneut ein Basis-RF-Modell auf Grundlage des ursprünglichen Datensatzes aufgesetzt, da die durch das Oversampling generierten Einträge das Ergebnis verfälschen würden.

Das folgende Balkendiagramm veranschaulicht das Ergebnis für die **zehn wichtigsten Inputvariablen.**

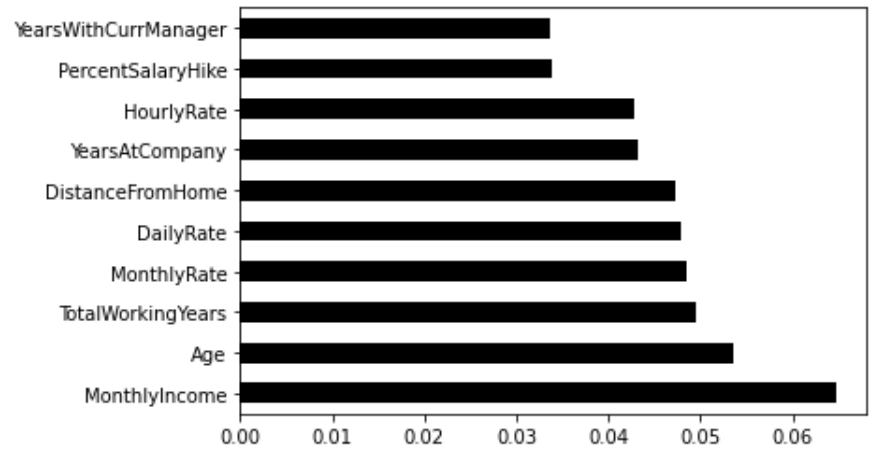

Abb. 14: Feature Importance (RF)

Anhand des Diagrammes ist erkennbar, dass das **fixe Monatsgehalt** mit Abstand den **größten Einfluss** auf den Unternehmensausstieg hat. Es folgen das Alter, die bereits geleisteten Arbeitsjahre sowie das gesamte Monatseinkommen, die die Person einnimmt.

5. Diskussion des Ergebnisses

Der Random Forest Algorithmus erzielt nach dem Hyperparameter Tuning eine Accuracy von **92,28 Prozent** und ist damit wesentlich besser als der KNN-Algorithmus. Eine weitere Reduktion der Features könnte zu einem besseren Ergebnis des KNN-Algorithmus beitragen, da trotz der durchgeführten Reduktion von 34 unabhängigen Variablen auf 26 unabhängige Variablen immer noch eine hohe Zahl an Features vorhanden ist. So könnte man in einem nächsten Schritt die numerischen Variablen auf ihren Einfluss auf die abhängige Variable prüfen und die Features ohne Einfluss aus dem Datensatz entfernen. Eine weitere Möglichkeit bestände darin, stark korrelierende numerische Attribute zu identifizieren (siehe Korrelationsmatrix im Notebook) und eins der zwei stark korrelierenden Attribute zu entfernen oder Programme zur Feature Selection anzuwenden. Für das vorliegende Modell wurde das Standardmaß der Euklidischen Distanz verwendet. Jedoch gibt es weitere Distanzmaßen, deren Ergebnisse mit dem der Euklidischen Distanz verglichen werden könnte, um das Distanzmaß mit dem besten Ergebnis zu identifizieren.

Darüber hinaus stellt sich die Frage, welche Metrik sich für das Projekt und die Fragestellung am besten eignet. Die Accuracy eignet sich, wenn die True Positives und True Negatives bedeutend sind und der F1-Score wenn die False Positives und False Negatives kritisch sind. Betrachtet man im vorliegenden Fall die False Positives, also die Mitarbeitenden, die fälschlicherweise als bleibend klassifiziert werden, so könnten diese kontextabhängig als kritisch eigestuft werden, da eventuell Ressourcen für diese Mitarbeitenden aufgewendet werden, um sie an das Unternehmen zu binden, obwohl diese das Unternehmen nicht verlassen werden. In diesem Fall würde die Wahl der Metrik auf den F1-Score fallen. Darüber hinaus eignet sich der F1-Score für unausgeglichene Datensätze besser als die Accuracy. Da bei Datensätzen mit echten Mitarbeiterdaten auch mit einer Unausgeglichenheit der Klassen gerechnet werden muss, würde sich aus der Argumentation heraus der F1-Score eignen. Geht man im vorliegenden Fall von unkritischen False Negatives und False Positives aus, so stellt die Accuracy nach der Durchführung des Oversampling ebenfalls eine valide Metrik dar. Dies ist jedoch kontextabhängig.

Durch die Anwendung weiterführender Algorithmen und die Gegenüberstellung aller Ergebnisse, könnte der Algorithmus mit den besten Ergebnissen ermittelt werden. Dies wäre im Praxisfall von großer Bedeutung. Für weiterführende Analysen wäre es außerdem interessant, wenn der Datensatz einen zeitlichen Anhaltpunkt für den Unternehmensaustritt bieten würde. So könnten die Unternehmensaustritte in einen zeitlichen Kontext gesetzt werden. Erste Analysen von möglichen Zusammenhängen wie beispielsweise zwischen dem Alter und dem Unternehmensaustritt sind mit dem gegebenen Datensatz allerdings schon möglich (siehe Abb. 13: Feature Importance (RF)).

Links:
Datensatz auf Kaggle:
https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset
GitHub Repository: https://github.com/LauraMorlok/Machine-Learning-Project
Jupyter Notebook: https://colab.research.google.com/drive/1TUn3-rlbyV_S0frR26LkDLNBAc2DNOEE?usp=sharing