

Spaceship Titanic

Bericht

für die Prüfung zum

Machine Learning Fundamentals

des Studiengangs Wirtschaftsinformatik – Data Science

an der

Dualen Hochschule Baden-Württemberg Lörrach

Presian Petrov und Jan Lade

19. Juni 2023

|  |  |
| --- | --- |
| Kurs | WDS21A |

Inhaltsverzeichnis

[Inhaltsverzeichnis V](#_Toc137651907)

[Abbildungsverzeichnis VIII](#_Toc137651908)

[Tabellenverzeichnis 9](#_Toc137651909)

[1 Einleitung 1](#_Toc137651910)

[1.1 Hintergrund 1](#_Toc137651911)

[1.2 Bedeutung 1](#_Toc137651912)

[2 Arbeitsumgebung 2](#_Toc137651913)

[2.1 Verwendete Software und Tools 2](#_Toc137651914)

[2.2 Hardwareanforderungen 2](#_Toc137651915)

[3 Datenbeschaffung 3](#_Toc137651916)

[3.1 Quelle der Daten 3](#_Toc137651917)

[3.2 Umfang und Struktur der Daten 3](#_Toc137651918)

[3.3 Wichtige Features und deren Bedetung 3](#_Toc137651919)

[4 Erkunden und Visualisieren 4](#_Toc137651920)

[4.1 Die erste Überschrift im vierten Kapitel (Ebene 2) 4](#_Toc137651921)

[4.2 Die zweite Überschrift im vierten Kapitel (Ebene 2) 4](#_Toc137651922)

[5 Data Preparation 5](#_Toc137651923)

[5.1 Behandlung von fehlenden Daten 5](#_Toc137651924)

[5.2 Datenbereinigung und Ausreißererkennung 5](#_Toc137651925)

[6 Preprocessing 6](#_Toc137651926)

[6.1 Skalierung der Daten 6](#_Toc137651927)

[6.2 Aufteilung des Datensatzes in Trainings- und Testdaten 6](#_Toc137651928)

[7 Modelling 7](#_Toc137651929)

[7.1 Vorgestellte ML-Verfahren 7](#_Toc137651930)

[7.2 Auswahl des passenden Modells 7](#_Toc137651931)

[7.3 Evaluirierung der Modelle mittels Metriken 7](#_Toc137651932)

[8 Erfahrungen 8](#_Toc137651933)

Abbildungsverzeichnis

[Abbildung 1: Hier steht die Beschriftung der Abbildung 1](file:///C:\Users\gerspachr\Documents\Word-Vorlage-Master\PA_BA_Arbeiten_Mastervorlage_V1.0.dotm#_Toc387246645)

# Einleitung

## Hintergrund

Der Bericht beschäftigt sich mit der Aufgabe, wo es darum geht, vorherzusagen, ob ein Passagier während der Kollision des Raumschiffs Titanic mit der Raumzeitanomalie in einer alternativen Dimension transportiert wurde oder nicht. Die Rettungsmannschaften haben die Herausforderung die Passagieren zu identifizieren und letztendlich wiederzufinden.

Das Ziel des Berichts ist die Aufzeichnungen zu analysieren und vorherzusagen welche Passagiere betroffen sind. Hierzu werden verschieden Datenanalyse und Maschinelles Lernen Algorithmen und Methoden verwendet.

## Bedeutung

Dieser Bericht bietet einen umfassenden Einblick in die Datenanalyse, Modelierung und vor allem die Evaluation der Ergebnisse und es richtet sich sowohl an Fachleute und aber auch an Personen, die kein großes Vorwissen in dem Bereich haben. Durch die gewonnenen Erkenntnisse soll der Leser einen Einblick erhalten und die Bedeutung von dem maschinellen Lernen verstehen.

# Arbeitsumgebung

## Verwendete Software und Tools

Um die Aufgabe zu lösen, wurden verschiedene Software-Tools für den Code eingerichtet, welcher in Python ausgeführt wird. Als erstes wurde eine virtuelle Umgebung mit dem Namen „myenv“ erstellt. Diese wurde danach aktiviert, damit man darin arbeiten kann. Dadurch beschränken sich alle zukünftige Python oder Paket Installationen. Um eine Datenanalyse führen zu können sowie die Funktionen für das maschinelle Lernen, wurden mehrere Python-Bibliotheken installiert. Diese sind:

* Numpy: für numerische Berechnungen und co.
* Pandas: zur Datenanalyse
* Matplotlib: Datenvisualisierungen erstellen
* Seaborn: erweiterte Visualisierung von statistischen Grafiken
* Scikit-learn: Bibliothek für das maschinelle Lernen und Datenanalyse
* Time: für Zeitmessungen
* Warnings: verwalten von Warnmeldungen

Für die Modellierung später im Bericht wurden mehrere Funktionen aus scikit-learn importiert, die dafür da sind, um die verschiedenen ML-Methoden anwenden zu können.

# Datenbeschaffung

## Quelle sowie Struktur der Daten

Für die Aufgabe wurde ein Datensatz aus Kaggle benutzt. Der Datensatz besteht aus mehreren Dateien, die die persönlichen Aufzeichnungen enthalten. Diese sind:

* Train.csv : Darin sind die Aufzeichnungen von circa 2/3 der Passagiere (ungefähr 8700) und werden als Trainingsdatensatz verwendet. Darin sind Informationen wie eine eindeutige ID, Heimatplanetet, ob der Passagier ein Kryoschlaf macht, Kabinennummer, Ziel, Alter, ob ein VIP-Service vorhanden ist und die Ausgaben.
* Test.csv : Darin sind die restlichen 1/3 Aufzeichnungen und werden als Testdatensatz verwendet.
* Sample\_submission.csv: Beispiel für die Einreichung der Vorhersagen. Darin sind die ID und zuvorsagende Variable, die „Transported“ heißt.

## Umfang und Struktur der Daten

Um mit der Analyse zu beginnen, müssen die 3 Dateien geladen werden. Danach kann ein Überblick geschaffen werden. Um zu überprüfen, ob Daten, bzw. Werte fehlen, wird die Funktion missing\_values\_table verwendet, wo man einen Überblick über die Anzahl der fehlende Wert gibt, und zwar in jeder Spalte. In den train und test Dateien sind auf jeden Fall fehlende Werte zu sehen und diese müssen behandelt werden, damit die Modellierung später richtige Vorhersagen geben kann. Zusätzlich wurde nach Duplikaten überprüft und darin sind keine zu sehen.

# Erkunden und Visualisieren

## Datenanalyse

Zunächst wurde die Verteilung der Zielgröße betrachtet, ob ein Passagier in eine andere Dimension transportiert wurde oder nicht. Dafür wurde ein Kreisdiagramm geplottet:

Ein Bild, das Text, Screenshot, Kreis, Diagramm enthält.

Automatisch generierte Beschreibung

Abbildung : Zielverteilung – Target Distribution

Das Diagramm zeigt ein recht ausgewogenes Ergebnis. Daher muss keine spezielle Technik angewendet werden, um mit den Ungleichgewichten umgehen zu müssen. Als weiterer Schritt wurde das Merkmal Alter analysiert, und zwar mit einem Histogramm. Hierbei ist die Altersverteilung der Passagiere dargestellt.

Ein Bild, das Screenshot, Text, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Abbildung - Altersverteilung - Age Distribution

Die Analyse ergab die Erkenntnisse, dass die Kinder (0-18 Jahre) eine höhere Wahrscheinlichkeit hatten in eine andere Dimension transportiert zu werden, während die jungen Erwachsenen (18-25 Jahre) eine geringere hatten. Für die Erwachsenen (über 25) ist die Wahrscheinlichkeit wie bei den nicht transportierten. Hierbei könnte ein neues Feature erstellt werden, das anzeigt, ob der Passant ein Kind, ein Jugendlicher oder ein Erwachsener ist.

Als nächstes wurden die kategorischen Features analysiert, welche folgende Erkenntnisse lieferten: Das VIP scheint nicht so nützlich zu sein, da die Verteilung relativ gliech ist. Das Feature CryoSleep liefert versprechende Ergebnisse. Bei der Analyse der qualitativen Features (PassengerId, Cabin und Name) ergaben sich die Erkenntnisse, dass die Passenger\_ID die Form gggg\_pp hat, wobei gggg die Gruppe angibt, mit der der Passagier reist, und pp die Nummer innerhalb der Gruppe ist. Die Kabine hat das Format Deck/Nummer/Seite, wobei Seite entweder P für Backboard oder S für Steuerboard stehen kann. Es könnte in diesem Fall sinnvoll sein die Gruppen aus dem Passenger Feature zu extrahieren, sowie Deck, Nummer und Seite aus dem Kabine Feature. Hilfreich in diesem Fall ist die Extrahierung von den Namen aus dem Namen Feature, um besser die Familien zu identifizieren.

# Data Preparation

In diesem Kapitel werden die Daten vorbereitet und berinigt und zwar für die Modellierung. Dieser Vorgang dient dazu, um eben eine richtige Vorhersage erstellen zu können.

## Feature Engineering

Als erstes wurde das Alter Feature genauer betrachtet und um die fehlende Werte zu behandeln, wurden Gruppen erstellt. Dabei sind sechs Gruppen entstanden: „Age\_0-12“ für Kinder bis „Age\_51+“, siehe Abbildung:

Ein Bild, das Screenshot, Text, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Abbildung - Verbesserte Age Group Distribution

Die Verteilung in diesem Fall wird eine gewisse, bzw. interessante Unterschiede zeigen, dass die Wahrscheinlichkeit angeht, transportiert zu werden.

Ein weiterer Aspekt, sind die Ausgaben. Dafür sind die Merkmale „RoomService“, „FoodCourt“, „ShoppingMall“, ,,Spa” und ,,VRDeck” zu betrachten. Diese werden im Code summiert und in ein neues Feature „Expenditure“ reingemacht. Darin sind die Gesamtausgaben jedes Passagiers zu sehen. Als Nächstes wurden zwei neue Features erstellt, um die Gruppenzugehörigkeit der Passagiere zu identifizieren. Die Gruppen heiße „Group“ und „Group\_size“. Durch diese Verteilung konnte festgestellt werden, dass die Passagiere, die alleine reisen, eine geringere Wahrscheinlichkeit haben, um transportiert zu werden in Vergleich zu den Passagieren, die in Gruppen sind. Das Problem in diesem Fall ist, dass das Feature „Group“ sich nicht als nützlich ergibt, aufgrund der zu großen Kardinalität von 6217 und das würde dem One-Hot-Codierung nicht helfen und Rahmen sprengen. Auf der anderen Seite ist das Feature „Group\_size“ gut und es wurde so komprimiert, indem eine Solospalte erstellt wurde. Darin wird erfasst, ob ein Passagier allein reist oder nicht. Zu dem Feature mit den Kabinen wurden die jeweiligen Merkmale extrahiert und dazu die fehlenden Werten behandelt. Anschließend wurden neue Merkmale erstellt, um die Kabinennummer in verschiedenen Regionen einzuteilen, welche analysiert werden können. Das wurde deshalb gemacht, weil festgestellt wurde, dass die Kabinennummer in Gruppen von jeweils 300 Kabinen unterteilt ist. Durch die Kategorisierung könnte festgestellt werden, in welchem Bereich sich der jeweilige Passagier befindet. Als letzter Schritt wurden die Familien größen anhand des Nachnamens berechnet. Dafür wurden die Nachnamen aus dem Feature „Name“ extrahiert.

## Fehlende Daten behandeln

Um fehlende Daten zu behandeln, haben wir die Trainings- und Testdatensätze zusammen gemacht, um es einfacher zu machen. Der einfachste Weg mit den fehlenden Werten umzugehen, ist die Verwendung des Medians für kontinuierliche Merkmale und des Modus (Modalwert) für kategoriale Merkmale. Das funktioniert gut genug, aber wenn die Genauigkeit bei dem Modellieren erreicht werden soll, muss nach Mustern innerhalb der fehlenden Daten suchen. Das funktioniert, indem die gemeinsame Verteilung der Merkmale „joint distribution“ betrachtet wird, d.h. ob Passagiere aus derselben Gruppe tendenziell aus derselben Familie stammen.

Nach all den Schritten ergaben sich keine fehlende Werte mehr und die Daten wurden wieder in Trainings- und Testdatensätze verteilt.

# Preprocessing

## Log transform

In diesem Schritt wurden die Überflüssigen Features entfernt. Damit sind die qualitativen, redundanten und kollinearen Features mit hoher Kardinalität gemeint. Als Nächstes wurde eine Logarithmentransformation benutzt, um die sogenannte Schiefe in den Verteilungen zu verringern. Hierbei werden auch die „Ausreißer“ reduziert. Das ganze wurde auf das Ausgaben Feature angewendet, da es von den Ausreißern stark verzerrt war. Zum Schluss wurden die Ergebnisse geplottet, bzw. grafisch dargestellt und die Transformation letztendlich auf die relevanten Features angewendet.

## Kodierung und Skalierung

In diesem Schritt wurden einige Schritte im Code durchgeführt. Dabei werden die numerischen (int64 und float64) und kategorischen (object) Spalten identifiziert und aufgeteilt. Danach wurden die numerischen Daten skaliert, indem den Mittelwert subtrahiert und durch die Standardabweichung geteilt wird. Bei den kategorischen Spalten wurde das One-Hot-Encoding angewendet. Dabei wird für jede Spalte eine neue binäre Spalte erstellt, die den Wert 1 aufweist, wenn die Kategorie passt, ansonsten 0. Als nächstes wurde die numerischen mit der kategorischen Transformation zusammengeführt mit dem Namen „num“ und „cat“. Diese Vorverarbeitung wurde mithilfe der Methode „fit\_transoform()“ angewendet um die daten zu transformieren. Durch all diesen Schritten wurden die Daten in einem passenden Format für Modellieren vorbereitet und könnte verwendet werden, um das Modell zu trainieren.

## Aufteilung des Datensatzes in Trainings- und Testdaten

Nach der Vorverarbeitung wurde eine Trainings-validierungs-Verteilung gemacht. Hierbei sind 80 % für das Training und 20 % für die Validierung zu verwenden.

# Modelling

## Vorgestellte und Auswahl ML-Verfahren

In diesem Abschnitt wird ein Dictionary erstellt, wo drei ausgewählte ML-Methoden sind:

* Logistic Regression: ein statistisches Verfahren, welches verwendet wird, um ein kategoriales Kriterium vorherzusagen. Dieses basiert auf der logistischen Funktion und ist gut für multiklassenbezogene Probleme.
* Support Vector Machine (SVM): statistisches Verfahren zur Klassifizierung von Objekten, bzw. deren Klassen zuzuordnen. Hier wird sozusagen versucht eine optimale Trennung zwischen den Klassen zu finden.
* K-Nearest Neighbors (KNN): Algorithmus, wo die Klassenzugehörigkeit basierend auf den k nächsten Nachbarn in den Daten bestimmt wird.

Zusätzlich wurden Parametergitter für eine Grid Suche erstellt, damit den Modellen mit verschiedenen Hyperparameterkombinationen trainieren zu können. Damit werden die besten Modelle ausgewertet. Mit der Validierungs wahrscheinlichkeit sind folgende Ergebnisse rausgekommen: für die logistische Regression wurde eine Validierungsgenauigkeit von 0,77 (=77%) und eine Trainingszeit von 0,20 Minuten erzielt, für SVM sind die Werte bei 0,79 und Trainingszeit von 10,20 Minuten. Für KNN sind die Werte bei 0,74 und eine Trainingszeit von 0,22 Minuten. Durch ein erstelltes Dictionary „best\_classifiers“ konnte eine „Liste“ mit den besten Klassifikatoren für jede der drei Modellen, welche auf den besten gefundenen Parametern nach der Grid-Suche erstellt werden. Die Klassifikatoren dienen dazu, um Vorhersagen für neue Daten zu treffen. Sie werden auf den transformierten und vorverarbeiteten Daten angewendet.

## Evaluierung der Modelle mittels Metriken

Wie bereits erwähnt werden die besten Klassifikatoren verwendet, um Vorhersagen zu erstellen mit den folgenden Evaluationsmetriken:

* Der F1-Score als eine Metrik, die eine Balance zwischen Präzision und Recall (Rückruf) misst. Hier wird angegeben, wie gut der Klassifikator ist. Zusätzlich wird identifiziert, ob es um positive oder negative Fälle handelt. Würde der Wert bei 1 liegen, dann wird von einer perfekten Präzision gesprochen, 0 entspricht in dem Fall für eine schlechte Leistung. Die Ergebnisse hierbei sehen jeweils wie folgt aus: Logistische Regression beträgt 0,777, SVC – 0,796 und KNN – 0,735. Diese Werte deuten auf eine gute bis akzeptable Balance zwischen Präzision und Rückruf.
* Die Matthews-Korrelationskoeffizient (MCC) ist dafür da, um die binäre Klassifikation zu bewerten. Hierbei werden Werte zwischen -1 (widersprüchliche Vorhersage) und 1 (perfekte Vorhersage) genommen. Die Werte liegen jeweils wie folgt vor: Logistische Regression – 0,543, SVC – 0,584 und KNN – 0,496. Diese Werte deuten auf eine gute bis durchschnittliche Leistung.
* Die Verwirrungsmatrix (Confusion Matrix) wird benutzt, um die Anzahl der korrekten und fehlerhaften Vorhersagen des Klassifikators für jede Klasse zu visualisieren. Die Matrix bietet dadurch eine gute Übersicht über die Leistung der Klassifikatoren.

## Kreuzvalidierung

Die Kreuzvalidierung ist eine Methode, womit die Leistung des Modells bewertet wird. Hierbei wird auf der Überanpassung (Overfitting) vermieden. Das Modell wird hier auf verschiedenen Teilmengen (in unserem Fall 10) in einer Schleife verteilt, trainiert und getestet. Am Ende wird die durchschnittliche Leistung über die Durchläufe berechnet. Dadurch erhält man eine zuverlässige Einschätzung der Modellleistung.

Die Ergebnisse sehen wie folgt aus:

* Die logitische Regression, gibt uns eine Validierung von 77,79 %
* Das Modell SVC erreicht die 80,12 %
* KNNs Wert liegt bei 77,23%.

Was heißt es für uns?

# Erfahrungen und Fazit

## Vergleich ziehen zwischen den Ergebnissen

## Eigene Erfahrungen Jan

## Eigene Erfahrungen Presian