# CAS Artificial Intelligence/Künstliche Intelligenz

# **Machine Learning Innovationsprojekt – Erfahrungsbericht**

## Verfasser: Marco Amrein und Jonas Furrer

## Datum: 03.02.2022

# Ziel der Arbeit

Dieses Innovationprojekt steht unter dem Titel «*Anwendung und Vergleich von Machine Learning und Deep Learning Algorithmen zur Vorhersage von COVID-19 Kennzahlen».* Das Ziel der Arbeit ist es also, Algorithmen aus dem Bereich des Machine Learning (ML) und Deep Learning (DL) zu entwickeln, um damit eine Vorhersage über die aktuellen und zukünftigen COVID-19 Kennzahlen zu treffen. Wir beschränken uns im Rahmen dieses Projekts auf die Vorhersage der COVID-19 Fallzahlen.

Die Daten stammen vom Portal opendata.swiss, das vom Bundesamt für Statistik betrieben wird [Quelle: https://opendata.swiss/de/dataset/covid-19-schweiz].

# Auswahl der Modelle und Einführung in Machine Learing und Deep Learning

In der Anmeldung für das Projekt haben wir folgende Modelle angegeben, die wir untersuchen und miteinander vergleichen möchten. Hier soll kurz auf die einzelnen Modelle und ihre Verwendungen eingegangen werden. Die Informationen zu dem Algorithmen stammen aus dem Buch von Sebastian Raschka und Vahid Mirjalili [Quelle zum Buch angeben].

*Machine Learning*

- Regression (linear, logistisch, polynomial)

- Support Vector Machine (SVM)

- Decision Tree

*Deep Learning*

- Multilayer Perceptron (MLP)

- Long short-term memory (LSTM)

- Convolutional Neural Network (CNN)

Nach der Diskussion mit den Dozierenden sind wir zum Schluss gekommen, nur eine Auswahl der Modelle zu implementieren. Dies aus folgenden Gründen:

1. Nicht alle oben beschriebenen Modelle sind für Regressions-Probleme (insbesondere mit Times Series Daten) geeignet. Sie kommen bei der Klassifizierung zum Einsatz.
2. Der Aufwand für die Implementierung und Evaluation aller Modelle ist sehr hoch. Mit der Beschränkung auf einige wenige Modelle können wir uns auf diese ausgewählten Modelle konzentrieren und diese genauer untersuchen.

Wir haben uns für folgende Algorithmen entschieden:

1. Polynomiale Regression
2. Convolutional Neural Network (CNN)
3. Long short-term memory (LSTM)

# Hypothese

Wir vergleichen die Modelle miteinander indem wir allen Modellen die gleichen Testdaten zur Verfügung stellen und die Resultate vergleichen. Als Testdaten nehmen wir die letzten 30% der gesamten Daten. Wir gehen davon aus dass die lineare polynomale Regression schlechter ist als die Neuronalen Netzwerke.

# Arbeitsaufteilung

Marco hat die Daten von covid19.admin.ch heruntergeladen[Todo:Jonas]. Dort sind auch die Features beschrieben [Todo:Jonas]. Im ersten Schritt wurden die Features von den jeweiligen CSV Dateien ausgewählt und in einer CSV Datei all\_data.csv zusammengeführt. Jonas hat dann im Preprocessing die Datentypen bereinigt, Werte bereinigt, aufgefüllt und die Datasets erstellt.

Diese Datasets wurden dann in den Modellen verwendetvon der Polynomialen Regression durch Marco, CNN und LSTM Model durch Jonas für das Training, Test und Verify verwendet.

covid19.admin.ch

Features auswählen

Preprocessing

Polynomiale Regression

CNM

LSTM

# Methodik

Als Collaboration Plattform für den Source Code haben wir uns für Git Hub [https://github.com/maggi71/ml-corona] entschieden. Durch das gemeinsame Repository waren Änderungen transparent sichtbar. Wir führten gegenseitig Reviews durch und konnten dadurch den Code verbessern. Lediglich Änderungen in Jupyter Notebook im Git sind nicht sehr gut nachvollziehbar, da nicht nur der Phyton Code eingecheckt wird, sondern auch die zuletzt ausgeführten Daten und Resultate, was zu vielen Differenzen führt.

Als Entwicklungsumgebung versuchten wir zuerst mit Google Colab [https://colab.research.google.com/] die Scripts aufzubauen. Dies war aber nicht sehr erfolgsversprechend. Weitere Tests mit Anakonda und Entwicklung im Browser brachten auch keine weiteren Verbesserungen.

Schlussendlich entschieden wir uns als Entwicklungsumgebung für Microsoft Visual Studio Code, da diese Entwicklungsumgebung viele nützliche Features wie Autovervollständigung und Refactoring integriert. Als Laufzeitumgebung wurde Anaconda verwendet als Remote Server in Visual Studio Code. Visual Studio Code ist sehr flexibel und einfach einzusetzen.

# Herausforderungen

Die erste Herausforderung war sicher mal Daten zu finden. Wir wollten eigentlich Kundendaten von CSS Kunden klassifizieren. Es zeigte sich aber dass die Datenschutz und Sicherheitshürden zu gross und zu aufwändig sind.

Aus diesem Grund entschieden wir uns für öffentliche Daten und luden die Covid 19 Daten vom Bund herunter.

Die Datenaufbereitung gestaltete sich als aufwändiger als geplant, bzw. uns wurden durch die erhöhten Aufwände klar wie wichtig gute Daten sind.

Das Dependency Management und die Laufzeitumgebung in den Gripp zu bekommen war aufwändig. Ein transparentes Dependency Management analog Java Maven oder Javascript NPM fehlt. Auch die Laufzeitumgebung ist nicht konfigurativ definiert sondern kann individuell ausgewählt werden was zu zusätzlichem Abstimmungsaufwand führte.

Optimierung der Hyperparameter

# Resultate

Der Aufwand für das Preprocessing ist um einiges höher als angedacht. 80% der Zeit verwendeten wir für das Preprocessing, 20% für das Erstellen der Modelle und das Lernen und optimieren.

Gute Daten sind das A und O für das Erstellen performanter Modelle. Die Aufwände Daten zu bereinigen haben wir unterschätzt. Zudem merkten wir dass eine dauernde Absprache notwendig ist, damit Beide die Daten gleich verstehen bzw. gleich interpretieren.

TODO:Erkenntnisse über den Vergleich der Modelle

TODO: Erkenntnisse über das erstellen der Modelle

Das Splitten der Times Series Daten gestaltete sich nach dem lesen des Beitrages von Keita Miyaki [https://medium.com/keita-starts-data-science/time-series-split-with-scikit-learn-74f5be38489e] einfacher und klarer.

# Persönliche Erfahrungen

Ein gutes Verständnis der Daten ist sehr wichtig. Die Bedeutung muss klar sein. Die Daten gemeinsam aufzubereiten und zur Verfügung stellen nahm mehr Zeit in Anspruch als geplant. Dank einer guten Beschreibung seitens Bund [https://www.covid19.admin.ch/api/data/documentation] war es uns beiden möglich die Daten zu verstehen.

# Fazit

Öffentliches ML oder DL Modell für jeden einsehbar und nachvollziehbar für Versierte. Modell kann auch jeder bei sich zu Hause ausführen. Das führt zu einer höheren Transparenz gegenüber der Bevölkerung. Durch den Aufbau einer Community könnte das Modell kontinuierlich verbessert werden.

# Literaturverzeichnis

## Auswahl der Modelle

Rückmeldung von Ladan Pooyan-Weihs zu den Modellen:

> ML- Linear Regression ist für unabhängige Daten (im Sinne der War’keit). Aber Time series sind abhängig.

> ML-Logistics: Dieses ist für die Klassifizierung. Es scheint nicht geeignet für dieses Projekt zu sein.

> ML-Support Vector Machine: hauptsächlich für die Klassifikation aber Regression ist auch möglich. Falls dieses untersucht werden sollte, könnte man dann sagen, wie gut dieses Model für Regression geeignet ist

> ML- Decision Tree: hauptsächlich für die Klassifikation aber Regression ist auch möglich. Falls dieses untersucht werden sollte, könnte man dann sagen, wie gut dieses Model für Regression geeignet ist

> DL- Multilayer Perceptron (MLP): Klassisch. Sie können dieses Model mit den Daten untersuchen und dann feststellen, wie gut dieses Model ist

> DL- Long short-term memory (LSTM): Es sollte mit Time series Date gehen. Falls dieses untersucht werden sollte, könnte man dann sagen, wie gut dieses Model für Regression geeignet ist

> DL- Convolutional Neural Network (CNN): Eher geeignet für Bilder als andere Dinge.

Wir beschränken uns auf folgende Modelle:

- Polynomiale Regression

- CNN

- LSTM