Stromerzeugung bei unterschiedlichen Wetterlagen – Datenaufbereitung und Analyse zur Korrelation Mithilfe von Regressionsalgorithmen in Weka

Korbinian Eller, Kay Gietenbruch

Abstract

In dieser Ausarbeitung wird das Vorgehen beim Sammeln von Daten, deren Aufbereitung und die Analyse mithilfe der Regressionsalgorithmen in Weka erläutert und die Ergebnisse dokumentiert.

Ziel ist es durch Wetterdaten des Deutschen Wetter Dienstes (DWD) auf Stromerzeugungszahlen der erneuerbaren Energien Solar und Wind (größte Beiinflussung durch Wetter) zu schließen

Idee

Zur gegebenen Themenstellung "Einen Themenbereich der KI vertiefen" war eine erste Idee die Klassifikation von Datensätzen. Dies war sehr ähnlich einer der letzten Aufgabenstellungen der Übungsstunden des Fachs. Es war vor allem aufgrund der Implementierung in der Programmiersprache C und damit dem greifbarmachen des Algorithmus interessant.

Der Gedanke der Klassifikation war schnell gefestigt nun fehlten noch die Daten mit denen gearbeitet werden soll. Über Daten wie Covid-19 Erkrankungs Zahlen, Bußgeldbescheide, Denkmalstandorte oder Amazon Personen Daten ist eine Idee herausgestochen.

"Es wäre doch interessant, Wetterdaten und Stromdaten in korrelation miteinander zu bringen und so die Stromerzeugung anhand des vorherrschenden Wetters deuten zu können". Und das war dann das Noema mit dem fortgefahren werden sollte. Ziel ist eine repräsentative Menge der Daten zu sammeln um eine Klassifikation sinnvoll ausführen zu können. Trotz alle dem war die Beschränkung der Daten auch ausschlaggebend. Festgelegt wurde sich dann auf den Zeitraum eines Jahres und wegen der Zeit in der die Daten gesammelt wurden (Ende 2023) war das Jahr 2022 passend für die Aufgabe.

Das Projekt konnte nun in mehrere Schritte unterteilt werden: die Daten sammeln, die Daten aufbereiten, den Klassifikator programmieren, Testen und verbessern und die Arbeit zu dokumentieren.

Nach Rücksprache mit dem Betreuer des Projekts Herrn Prof. Dr. Baumann wurde jedoch klar, dass eine Klassifizierung der Daten nicht das geeignetste Modell für die Analyse der Korrelation ist. So wurde der Plan neu geschrieben und eine Analyse mithilfe von Regressionsalgorithmen in dem Machine-Learning Programm Weka stand ab dem Zeitpunkt im Vordergrund. Dafür sollten die Daten noch angepasst und dann mit Weka und den Regressionsalgorithmen experimentiert werden. In dem Sinne, dass das am best geeignete Modell zu Regression gefunden wird!

Daten Recherche

Die Daten, mit denen die Regression in Weka betrieben werden soll, müssen zuerst zusammengetragen werden. Sowohl die Daten des DWDs, respektive die Wetterdaten, als auch die der Bundesnetzagentur(BNetzA), welche die Daten der Stromerzeugung bereitstellt, stehen leider nicht auf der jeweiligen Websites als Download in Form von z.B. ".csv" oder ".xls" Dateiformaten zur Verfügung. Hier war also eine andere Herangehensweise gefragt. Nun muss zwischen den Daten des DWD und denen der BNetzA unterschieden werden, da diese in unterschiedlicher Form vorliegen und somit auch einen unterschiedlichen Sammel-Prozess unterlaufen sind.

Es ist anzumerken, dass dieser Arbeit einige Dateien beigelegt sind unter denen sich auch ".md" Dateien befinden in welchen sehr speziell auf die einzelnen Schritte und die Struktur eingegangen wurde.

BNetzA Daten

Die Bundesnetzagentur stellt ihre recht simplen Daten (Im vergleich zum DWD) auf der Website "Strommarktdaten" sehr anschaulich in einem Flächendiagramm dar. Aus solch einem Diagramm Werte in eine Datenbank zu übertragen wäre aber bei den stündlichen Werten von dem ganzen Jahr 2022, sprich 8760 Zeilen einer CSV Datei und jeweils zwölf Datentypen undenkbar. Glücklicherweise liefert die BNetzA nicht eine Grafik an die Website, sondern die Stündlichen Daten in mehreren Dateien vom Typ ".json" an den Browser, wo dann ein Diagramm erstell wird.

Das heißt, durch einen einfachen "cURL" Befehl auf der Kommandozeile, kann eine solche Datei, wenn der vollständige Name bekannt ist, heruntergeladen werden. Die Namen bestehen aus einer Zahlenkodierung der unterschiedlichen Daten-Arten (z.B. Kernenergie - 1224, Wind Onshore - 4067), dem Kürzel DE für Deutschland, der

”Auflösung” der Daten (z.B. hour, day) und dem Epoch Zeitstempel für den frühesten Aufgezeichneten Wert in dieser Datei. Diese Dateien erfassen immer eine ganze Woche, das heißt die Epoch Zeit codes unterscheiden sich immer um genau 604.800.000 Millisekunden, da dies genau 7 Tage abbildet. So konnte das Jahr 2022 schnell in Epoch Codes abgebildet werden und es stand fest welche Dateien benötigt werden, da alle Type erzeugten Stroms geladen wurden, weil es relativ problemlos machbar war.

Der Aufbau der angesprochenen ”.json” Dateien ist ein Array mit 180 Datenpunkten, die jeweils zwei Werte haben. Einmal den Epoch Zeit code und die Höhe des gemessenen Stroms des jeweiligen Typs in MWh. Am Anfang der Datei beschreibt eine Versionsnummer und ein Zeitpunkt (ebenfalls Epoch) die Erstellung der Datei. Der Nächste Schritt war nun die Dateien in einem iterierendem Python Programm auf der eigenen Festplatte zu speichern. Einziges Problem hierbei war nur, dass die Dateien immer Montags um 0 Uhr beginnen, aber das Jahr 2022 nicht auf einen Montag begonnen bzw. auf einen Sonntag geendet ist. Folglich sind mehr Dateien, als es eigentlich braucht um ein Jahr abzubilden, geladen worden.

DWD - Daten

Der Deutsche Wetterdienst stellt seine Daten auf einem opendata Server zu Verfügung. Nachdem sich in der Dateistrucktur zurechtgefunden wurde, sind auch die ersten Probleme aufgekommen:

- Welche Wetterdaten sollen benutzt werden?
- Wie werden die gewünschten Zeiträume gefunden?
- Welche Dateien enthalten welche Daten?

Einiges das im ersten Moment sehr unklar erschien! Die Lösungen waren dann wiefolgt:

Datenart Es wurde sich auf Eigenschaften beschränkt die aus subjektiver und ungeschulter Sicht stark in die Stromproduktion als Faktoren miteinfließen. In erster Linie handelt es sich um erneuerbare Energien, also galt es bestimmte Naturphänomene, die Photovoltaik-Anlagen oder Windkraftanlagen beiinflussen, auszumachen. Dies sind:

- Lufttemperatur Feuchtigkeit
- Bedeckungsgrad des Himmels
- Niederschlag
- Sonnenscheindauer
- Sichtweite
- Wind

Das waren die Unterteilungen der Messwerte auf Seite des DWD

Zeiträume Da unter jeder der eben aufgezählten Sektionen in dem Dateisystem des Servers eine Unterteilung in ”recent” und ”historical” stattfand und der gewünschte Zeitraum das Jahr 2022 war, galt es herauszufinden wie die Aufteilung zustande kam. Leider gibt es keine klare Unterteilung und somit blieb nichts anders übrig als in beiden Ordnern nachzuschauen.

Dateien In den zeitunterteilenden Ordnern liegen genau eine Datei, die die Liste der Stationen mit ihren Eigenschaften aufzählt, welche den betrachteten Wert aufzeichnen sollen und hunderte komprimierte Ordner, benannt nach Stationscode, Daten der Erfassung (Nur in ”historical”, nicht in ”recent”) und erfasste Eigenschaft. Teilweise reichten die erfassten Daten von 1970 bis 2001 oder von mitte 2022 bis Anfang 2023, es war also kein System hinter den Aufzeichnungen zu Erkennen. Die ”.zip” Ordner enthalten mehrere Datein, teilweise ”.html” und teilweise ”.txt” Dateien und das sogar manchmal in doppelter Ausführung. trotz alledem gibt es in jedem Ordner eine ausschlaggebende ”.txt” Datei, die die vom Ordnernamen versprochenen Daten beinhalten.

In den angesprochenen Datein ist allerdings eine gewohnte Struktur wiederzufinden. Aufgebaut wie eine ”csv” Datei nur mit Semikola getrennte Spalten. Die erste Spalte besteht aus dem Datum gepaart mit der Stundenzahl des Tages im 24h-Format, zu welchem Zeitpunkt die Daten der Reihe von den Messinstrumenten ausgelesen wurden. Die nächsten Spalten (1-3, je nach Datei) bestehen aus den gewünschten Attributen wie Sonnenscheindauer. Das Ende der Zeile macht immer ein ”eor” als Zeilenende-Indikator. Manche Werte, unterschiedlich von Station zu Station und je nach Tageszeit sind gar nicht aufgelistet bzw. als fehlend (-999 in der Datei) gekennzeichnet. Für einige Station, wie zuvor schon thematisiert, stehen überhaupt keine Daten für bestimmte Messattribute in Form von Dateien in dem gewünschten Zeitraum zur Verfügung. Oft wurden auch nur in einem Teil des Jahres 2022 Daten erhoben, bzw. zur veröffentlichung auf dem Server freigegeben, was zu ”halben” Dateien führt.

Trotz all diesen Hürden war es klar, dass das sammeln dieser Daten nicht ”per Hand” realisierbar ist, sondern durch Automatisierung geschehen soll. Ohne vorkenntnisse in Python war das Vorhaben in der Tat anspruchsvoll, aber die überaus breit gefächerte Dokumentation der Sprache und den vielen Bibliotheken erleichterten das Programmieren sehr. Zu den Vorlesungen zu erscheinen, scheinte sich in diesem Zuge zu lohnen, als die Benutzung von RegEx (und somit den Grundlagen der Informatik) das Filtern der Datein um einiges erleichterte. Einmal geschrieben, iterierte das Skript durch alle oben genannten Datenarten und sammelte und entpackte die Dateien auf der eigenen Festplatte. Die Wetterdaten vollständig heruntergeladen blieb noch die Daten der Bundesnetzagentur.

Daten Aufbereitung

Der nächste Schritt ist die Datenaurbereitung, bei der die eben gesammelten Daten in vorzugsweise ”.csv” Dateien geschrieben/konvertiert werden.

Zuerst wurden zwei eigene Dateien erstellt die dann später noch raffiniert und letztendlich sogar partiell vereinigt wurden.

Stromdaten

Die Daten der Bundesnetzagentur tabellarisch zu sortieren war keine Kunst, da es glücklicherweise passende Python

Bibliotheken gibt, die ".json" Dateien in Listen einlesen können und erleichtern somit die Erstellung einer Datei im ".csv" Format.

Ein Beschneiden der Daten war nun noch zu erledigen, da wie oben genannt die Epoch codes nicht genau passten. Also wurden die Überflüssigen Codes (die vor dem 01.01.2022 und die nach dem 12.31.2022) abgeschnitten. Dann wurden die Epoch codes mit passender Bibliotheken in Daten übersetzt und eine "header"-Zeile wurd der Datei vorangesetzt. Durch die Epoch codes, konnte das Zeitumstellungssystem in Deutschland ignoriert werden, das die Unix-Zeit das nicht beachtet.

Wetterdaten

Eine andere Handhabung gebührt hier den Wetterdaten.

Verfeinerung

Die Verfeinerung

Regressionanalyse mit Weka - Kay

Vorgehen beschreiben. Welche Modelle? Weka Version? Welche Files? Gab es Probleme? Auffällige Zeiten? Kompatiblität mit fehlwerten? Hat was gespackt uws. Sachen aus dem Weka Buch hilfreich?

Vergleich von Modellen - Kay

Was war denn eignetlich des beste und warum. Was war das Fehlermaß? Wie haben sie sich Zeitlich geschlagen? Weka gut/schlecht? Andere Ideen oder so. Wie war eigentlich Klassifikation - hätte das doch Sinn ergeben?

Fazit

Was konnten wir aus dieser Arbeit schließen? Erfolg der Bearbeitung der Fragestellung? Aufwand/Zeit Verhältnis. Ergebnis der Modelle. Hätte man was besser machen können und wie?

References

Bundesnetzagentur. 2023. BNetzA (01.01.2022-12.31.2022). *https://www.smard.de/home/marktdaten*.

Deutscher-Wetterdienst. 2023. DWD (01.01.2022-12.31.2022). *https://www.smard.de/home/marktdaten*.

Witten, I. H.; Frank, E.; and Hall, M. A., eds. 2011. *Data Mining: Practical Machine Learning Tools and Techniques*. Burlington, MA: Morgan Kaufmann, third edition.

RAUSNEHMEN:

Zu Datagathering

Siehe Doku in den MD files.

Wetter: Von den Zip dateien in die CSV Dateien. Von den CSV In eine Große CSV. Fehlwerte, fehlende Csvs, erst einmal eine Große mit allen und dann eine kompaktere mit nur den 157 vollständigen. Dann einen noch kompaktere ohne RR-2 i think. Dann eine noch kompaktere meaned csv. -¿ ab in eine arff file für weka über den Weka explorer. Power: Viele einzelne Json Dateine die nicht über genau das Jahr auch gehen. epoch in normale time codes. Json in CSV umwandeln. Die Paar csvs dann in eine größere mit den Epoch timecodes. Dann auf das Jahr 2022 beschneiden mit den Epoch Codes