Data Mining & Grundlagen Maschinelles Lernen 1

Wintersemester 2024/25

Semesterprojekt: Solar Power Forecasting Challenge

1 Organisatorisches

1.1 Zweck und Scope

Im Modul Data Mining & Grundlagen Maschinelles Lernen 1 sollen Sie unter anderem lernen, mathematische Vorhersagemodelle für Klassifizierungs- oder Regressionsprobleme zu entwickeln und zu bewerten. Sie sollen dabei die Kenntnisse und Techniken, die im Laufe der Vorlesung vermittelt wurden, auf ein konkretes Problem anwenden, um ein Modell zu entwickeln, das auf *unbekannten* Daten möglichst gute Vorhersagen liefert.

Die einzelnen benötigten Techniken wurden im Laufe des Semesters bereits in kleineren Übungsprojekten angewendet. Die Lernziele der vorliegenden Aufgabe sind:

- Das Erlernte mit relativ wenigen Vorgaben für ein neues Problem einzusetzen und dabei erlerntes Wissen aus verschiedenen Einheiten zu verknüpfen. Die einzelnen Schritte sollen dabei nachvollziehbar dargestellt und begründet werden.
- Sauber strukturierten und kommentierten Code zu erzeugen, der von anderen Personen übernommen und ggf. weiterentwickelt werden könnte.

1.2 Organisation und Termine

Die Übung ist durch Projektgruppen bestehend aus **bis zu drei** Studierenden umzusetzen. Die Gruppen müssen in ILIAS registriert sein.

- **Start: 27.11.2024**
- **Abgabe Code, Vorhersagen und Aufgaben (Detailhinweise s.u.): 10.12.2024**
- **Abgabengespräche (Detailhinweise s.u.): 11.12.2024 und 12.12.2024**

Das Projekt wird basierend auf dem Code, der Vorhersage auf den Testdaten und der Bearbeitung der Aufgaben benotet. Die Projektnote macht **30% der Modulnote** aus. Das Bestehen des Projekts mit mindestens 4.0 ist Voraussetzung zur Teilnahme an der Klausur. Wenn Sie das Projekt in einem vorherigen Semester bereits bestanden haben, müssen Sie es nicht noch einmal bearbeiten.

1.3 Anforderungen und Abgaben

1.3.1 Code

Der Code sollte sauber strukturiert, nachvollziehbar und lauffähig sein (ggf. Hinweise auf verwendete Pakete und Versionsnummern). Es ist Ihnen überlassen, ob Sie den gesamten Code in einem einzigen Jupyter-Notebook implementieren oder ob Sie den Code in mehreren .py-Dateien organisieren. Benutzen Sie sowohl Kommentare im Code als auch Markdown Zellen um ihr Vorgehen zu erläutern. Achten Sie auch auf sinnvolle Bezeichner für die Variablen. Grundsätzlich gehe ich davon aus, dass sie die in der Vorlesung behandelten Pakete wie `scikit-learn` , `Pandas` und `NumPy` für die Analysen benutzen. Falls Sie gänzlich andere Pakete (oder Modelle) benutzen möchten, ist das grundsätzlich auch möglich. Sprechen Sie dies aber vorher mit mir ab.

Grundsätzlich gilt: Sie müssen bei der Vorstellung Ihrer Abgabe in der Lage sein, mir jede Zeile Ihres Codes zu erklären. Bedenken Sie das vor allem, wenn Sie sich dafür entscheiden, KI Tools zu verwenden.

1.3.2 Vorhersagen

Die Vorhersagen auf den beiden Testsets reichen Sie bitte im Competition Portal unter `http://193.196.38.142:8501` ein. Sie erhalten danach Rückmeldung über den root mean squared error (RMSE) Ihrer Vorhersagen auf Testset 1. Am Ende des Projekts wird der RMSE für Testset 1 und Testset 2 berechnet. Auf dessen Basis wird dann die finale Rangliste erstellt.

Den Gewinnerteams winken folgende Preise:

- 1. Platz: 10% Klausurpunkte als Bonus
- 2. Platz: 7% Klausurpunkte als Bonus
- 3. Platz: 5% Klausurpunkte als Bonus

Um Abgaben im Portal hochladen zu können, teilen Sie mir bitte einen **Teamnamen** mit. Sie erhalten dann von mir einen Teamkey.

1.3.3 Aufgaben

Erstellen Sie für die Aufgaben 1–4 einen kurzen Report als pdf, in der Sie ihre Erkenntnisse zusammenfassen (Plots, Begründung der Vorgehensweise). Für Aufgaben 5 und 6 reicht (kommentierter) Code.

1.3.4 Abgabengespräch

Ihre Abgaben stellen Sie mir in einem kurzen Gespräch vor, bei dem alle Teammitglieder anwesend sein müssen. In dem Gespräch werde ich Rückfragen zu den Ergebnissen und dem Code stellen. Jedes Team bekommt dafür von mir einen Termin (die Abgaben müssen vorher über ILIAS eingereicht werden).

2 Das Projekt

In Stromnetzen müssen Angebot durch Stromerzeuger und Nachfrage durch Stromverbraucher möglichst gut übereinstimmen, um die Stabilität des Stromnetzes zu gewährleisten. Wird zu wenig Strom eingespeist, kann es sein, dass der Bedarf nicht gedeckt wird – es käme also zu Stromausfällen. Wenn mehr Strom eingespeist wird als nachgefragt wird, kann es zu Instabilitäten und Schäden an der Netzinfrastruktur kommen. Der Stromhandel dient hier als primärer Mechanismus, um Angebot und Nachfrage auszubalancieren.

Für die Stromproduktion wird oft eine große Anzahl kleinerer Produzenten zu *virtuellen Kraftwerken* zusammengeschlossen. Ein virtuelles Kraftwerk kann dann an der Strombörse, an der Strom zu dynamischen Preisen gehandelt wird, teilnehmen. Zur Berechnnung des sog. *Day-Ahead-Preises* fordern die Stromnetzbetreiber von den Marktteilnehmern 24 Stunden im Voraus eine Prognose, wie viel Strom diese am nächsten Tag zu einer gewissen Stunde bereitstellen werden. Abweichungen von der Prognose werden anschließend durch einen Ausgleichsmechanismus beglichen, in dem je nach Höhe der Abweichungen für den Stromproduzenten Kosten anfallen können. Eine möglichst genaue Prognose der produzierten Strommenge zu erstellen zählt somit zur Schlüsselkompetenz eines Stromerzeugers mit vielen erneuerbaren Energien im Portfolio.

In diesem Projekt sollen Sie für ein virtuelles Kraftwerk mit einem großen Menge an Photovoltaikanlagen Modelle entwickeln, die stundengenaue Prognosen für die erzeugte Strommenge treffen. Für die Entwicklung des Modells muss der gesamte Machine Learning Workflow durchlaufen werden, von der Datenvorverarbeitung und -analyse über das Modelltraining verschiedener Modelle bis zur Auswertung der Modellgüte. Schließlich sollen auf zwei bereitgestellten Testsets Prognosen erzeugt werden, um die Generalisierung der Modelle zu testen.

2.1 Beschreibung der Daten

Für die Modellentwicklung stehen folgende Daten zur Verfügung:

- `energy_train.pkl`: Daten zu erzeugter Strommenge pro Stunde (Ënergiedaten").
- `forecasts.pkl`: Historische Wettervorhersagen, mit deren Hilfe man die erzeugte Strommenge pro Stunde vorhersagen kann ("Wettervorhersagen").

Eine Zeile der Datei `forecasts.pkl` repräsentiert eine Wettervorhersage zu einem bestimmten Zeitpunkt. Die Spalten haben folgende Bedeutung:

- `ref_datetime`: Zeitpunkt, an der die Wettervorhersage veröffentlicht wurde (Timestamp).
- `valid_time`: Stunde, für die die Wettervorhersage gültig war, angegeben in Stunden nach der Erstellung der Wettervorhersage. Beispiel: Wenn `ref_datetime` `2020-09-20 00:00:00+00:00` und als `valid_time` 27 angegeben ist, so gilt die Zeile, die diese Wettervorhersage repräsentiert, für den 21.09.2020, 3:00 bis 4:00 Uhr morgens (27 Stunden nach Erstellung der Vorhersagen).
- `SolarDownwardRadiation`: Vorhersage der Sonneneinstrahlung in W/m².
- `CloudCover`: Anteil der Stunde, in dem Bewölkung vorhergesagt ist.
- `Temperature`: Vorhersage der Temperatur in °C.
- `Weather Model`: Welches Wettervorhersagemodell verwendet wurde (DWD = Deutscher Wetterdienst, NCEP = National Centers for Environmental Prediction)

Eine Zeile der Datei `energy_train.pkl` repräsentiert die erzeugte Strommenge während einer Stunde. Die Spalten haben folgende Bedeutung:

- `dtm`: Beginn einer Stunde (Erhebungszeitraum)
- `ref_datetime`: Zeitpunkt der neuesten verfügbaren Wettervorhersage, die verwendet werden kann. Da die Strommenge 24h im Voraus vorhergesagt werden muss, liegt `ref_datetime`, also der Veröffentlichungszeitpunkt der Wettervorherage, mindestens 24 Stunden zurück.
- `Solar_capacity_mwp`: Menge der verfügbaren PV-Anlagen in mwp (Megawattpeak)
- `Solar_MWh`: Erzeugt Strommenge in Megawattstunden (das soll vorhergesagt werden).

Des Weiteren gibt es zwei Dateien `energy_test1.pkl` und `energy_test2.pkl` für die beiden unbekannten Testsets. In diesen fehlt das Label `Solar_MWh`.

2.2 Aufgaben

Aufgabe 1 (10 Punkte)

Untersuchen Sie die Energiedaten. Bearbeiten Sie dabei folgende Fragen bzw. Aufgaben:

1. Wie viele Datenpunkte, die vorhergesagt werden sollen, gibt es in den Trainings- bzw. Testdaten?
2. Visualisieren Sie den Tagesverlauf der Stromerzeugung für zufällig ausgewählte Tage, z.B. die Geburtstage der Teammitglieder. Beschreiben Sie die Kurven.
3. Visualisieren Sie den Gesamtverlauf der Stromerzeugung für den Zeitraum der Trainingsdaten. Beschreiben Sie die Kurve.

Aufgabe 2 (10 Punkte)

Führen Sie Energiedaten und Wettervorhersagen zusammen. Untersuchen sie Zusammenhänge zwischen dem Label und den Attributen der Wettervorhersage.

1. Erzeugen Sie Plots, aus denen man einen möglichen Zusammenhang zwischen Label und Attributen erkennen kann.

2. Welche Attribute lassen auf einen starken Zusammenhang schließen?

Aufgabe 3 (10 Punkte)

Führen Sie basierend aus den Analysen der vorherigen Aufgaben geeignete Vorverarbeitungsschritte durch, z.B. Behandlung von Ausreißern und fehlenden Werten, Skalierung der Daten.

Aufgabe 4 (20 Punkte)

Generieren Sie neue Features, die für die Vorhersage des Umsatzes aussagekräftig sein könnten.

Aufgabe 5 (30 Punkte)

Trainieren Sie drei verschiedene Modelle, die in der Vorlesung behandelt wurden: ein lineares Modell (einfache lineare Regression, Ridge, Lasso), einen Entscheidungsbaum und ein Ensemble-Modell (Gradient Boosting oder Random Forest)

1. Optimieren sie Hyperparameter der Modelle mittels Suche und Kreuzvalidierung (z.B. mit Hilfe der Klasse `GridSearchCV` aus `scikit-learn`). Überlegen Sie dazu zunächst (mit Hilfe der Vorlesungsunterlagen und der Dokumentation der Methoden in `scikit-learn`), was für die jeweiligen Modelle Hyperparameter sind und für welche sich eine Optimierung ggf. lohnen könnte.

2. Welches sind die wichtigsten Features für die jeweiligen Modelle?

Wählen Sie als Evaluierungsmetrik den root mean squared error (RMSE).

Aufgabe 6 (20 Punkte)

Erstellen Sie Vorhersagen für die Testdaten `energy_test1.pkl` und `energy_test2.pkl` mit dem besten Modell aus der vorherigen Aufgabe. Speichern Sie die Vorhersage in einer Spalte `Solar_MWh_pred`, die Sie dem DataFrame aus `energy_test1.pkl` bzw. `energy_test2.pkl` hinzufügen. Speichern Sie die DataFrames mit der Methode .to_pickle() in `.pkl` Dateien und laden Sie diese im Portal hoch. Es wird die Güte Ihrer Vorhersagen bewertet.

2.3 Tipps

- Planen Sie genug Zeit für das Datenverständnis und die Datenanalyse ein, bevor Sie mit Modelltraining und -evaluation beginnen. Greifen Sie dabei auch auf ihr Wissen aus anderen Veranstaltungen zurück, z.B. deskriptive Statistik oder Data Analytics & BI.

- Starten Sie mit einem sehr einfachen Modell, um eine Baseline zu bekommen, an der Sie sich orientieren können. Verfeinern Sie es danach Stück für Stück. Das hilft, Modellierungs- und Programmierfehler schneller ausfindig zu machen und Overfitting vorzubeugen.

- Bei der Erstellung neuer Features hilft es, sich zu überlegen, welche Information denn generell hilfreich sein könnte, den Output einer PV-Anlage zu einem bestimmten Zeitpunkt vorherzusagen. Anschließend überlegen Sie sich, wie Sie diese Information aus den Daten extrahieren können.

- Wenn Sie Daten im Portal abgeben, bekommen Sie als Rückmeldung den RMSE auf dem Testset 1. Erschrecken Sie nicht, wenn der Fehler deutlich höher ist als auf den Trainingsdaten. Die Daten neigen stark zum Overfitting und auch das beste Modell wird auf den Testdaten u.U. deutlich schlechter sein als auf den Trainingsdaten.
- Überlegen Sie sich frühzeitig, wie Sie die Validierungsdaten wählen, auf denen Sie Ihr Modell testen. Erinnerung: Das Ziel ist es nicht, die Validierungsdaten so zu wählen, dass das Modell darauf möglichst gut abschneidet, sondern so, dass Sie eine möglichst realistische Einschätzung der Performance des Modells auf unbekannten Daten geben (hier also Testset 1 und Testset 2).