

Read-Me zu Batteriespeicher Dashboard

Autoren: Mauro Stoffel, Marc-Alexander Iten

Codebeschreibung

Verwendete Packages

Paketname	Komponentenname	Version	Verwendet in
dash	dash dash-core-components dash-html- components dash-table	2.17.0 2.0.0 2.0.0 5.0.0	dashboard.py generateDiagramms.py
numpy	-	1.26.4	dashboard.py fetchData.py
dash_bootstrap_components	dash-bootstrap-components	1.6.0	dashboard.py
plotly	-	5.22.0	generateDiagramms.py
pandas	-	2.2.2	generateDiagramms.py fetchData.py
scipy	-	1.13.0	generateDiagramms.py

dashboard.py

Dies ist das Hauptskript, welches ausgeführt werden muss, um das Dashboard zu starten. Es erstellt das Dash App-Struktur. Dies beinhaltet die Inhalte der Dropdown Menüs, die «HTML» Struktur und die Callback-Funktionen. Dies wird alles im Skript selbst ausgeführt und ist nicht in Funktionen verpackt.

generateDiagramms.py

Dieses Skript befasst sich mit der Erstellung und Übergabe alles im Dashboard vorhandenen Plotly Diagrammen. Es wird zu Beginn beim Erstellen des Dashboards aufgerufen und danach nochmals in den Callback Funktionen. Die Funktionen grossVerbraucher(), batterieAnalyse(), PVerzeugung_Verbrauch(), generateCenterTable1(), generateCenterTable2(), generateCenterGauge() werden im Hauptskript dashboard.py importiert.

fetchData.py

Dieses Skript befasst sich mit dem Beziehen und Aufbereiten der durch die API zur Verfügung gestellten Daten. Es geht dabei hauptsächlich um das Beziehen der Stromdaten, welche in einem ersten Schritt in den korrekten Datentyp gecasted werden und danach noch durch zusätzliche Daten, welche für die Darstellung nötig sind, erweitert werden. Zusätzlich wird eine Liste von bekannten Batterietypen aus dem API bezogen und für die Darstellung im Dropdown Menü aufbereitet. Zuletzt wird in der generateBatData() Funktion noch der Batterieladeverlauf für die momentanen Einstellungen berechnet.

styling.css

Dies ist das Cascading-Stylesheet, welches für die Verbesserung der Visualisierung zuständig ist. Es wurde in die Unterkapitel General, Titel, Dropdowns, 1. Reihe, 2. Reihe, 3. Reihe und Tabs unterteilt, für bessere Leserlichkeit.

Benutzeranleitung des Dashboards

Installation und Aufstarten

1. Anlegung einer lokalen Kopie des GitHub Repositories.
2. Installation der im vorigen Kapitel erwähnten Package Versionen.
3. Ausführung des dashboard.py Skriptes und Navigation zu dem im Terminal angegebenen Link. Falls bei diesem Schritt Zugriffsprobleme auf die API bestehen, bitte ich euch mit Marc in Verbindung zu treten, dass er euch Zugriff auf die API gewähren kann.

Benutzung

Zu Beginn kann oben in den Dropdown-Menüs, die für die bedienende Person zutreffende Strompreis und Stromvergütung ausgewählt werden. Danach kann die Batterie, die untersucht werden will und die zu erwartende Batterieeffizienz ausgewählt werden. Nun wird unten zentral die Berechnung der Daten in der Tabelle angezeigt und dazu die Amortisationszeit, welche die gewählte Batterie in der momentan gewählten Konstellation hätte. Hier ist noch wichtig zu erwähnen, dass die zentrale Tabelle nicht durch die Auswahl des Zeitraumes beeinflusst wird. Die Zeile «gesparte Energie» wird direkt aus der API, also aus den Rohdaten geholt. Die restlichen Berechnungen in der Tabelle sind ebenfalls nicht von der Zeit abhängig.

Um die Validität der Daten zu prüfen, dienen die Grafiken rechts und links von der Tabelle. Rechts kann die Erzeugung und der Verbrauch während der gewählten Zeit untersucht werden. Links sind die beiden Balken «Tage mit Überproduktion» und «Tage mit voller Auslastung». Der Balken «Tage mit Überproduktion» dient als Kriterium des Standortes. Wenn zum Beispiel nicht genug Strom durch die Photovoltaikanlage generiert wird, um die den verbrauch überhaupt zu übertreffen, wäre eine Batterie gar nicht erst nötig. Der andere Balken «Tage mit voller Auslastung» dient zur Analyse der Batterie. Wenn dieser Balken bei jeder erdenklichen Auswahl an Zeitdaten immer auf 100% ist, wurde eine zu kleine Batterie gewählt. Wenn sie immer auf 0% ist, wurde eine zu grosse Batterie gewählt.

Über das Anklicken des Tabs «Batterie Simulation» können die aufgezeichneten Daten und die simuliert Batteriedaten, welche durch die Dropdown-Menüs beeinflusst werden, gesichtet werden.

Zwei Abschlussdetails:

- Das Impressum kann über das «i» Symbol im Titel geöffnet werden.
- Da ich weiss, dass es bemängelt wird, möchte ich klarstellen, dass horizontale Achsenbeschriftungen bei den y-Achsen der Plots nicht möglich sind. Dies wurde bestätigt durch einen «Co-Founder» der Plotly Bibliothek. (Siehe Anhang)

Wichtigste Programmierschritte

Vor Besprechung

- Erstellung der einzelnen Plots via Plotly
- Erstellung der Grundstruktur und des generellen Aufbaus des Dashboards
- Implementierung der Diagramme in das Dashboard
- Erstellung der mittleren Tabelle mit Pseudodaten
- Erstellung der Callbacks
 - o Wurden aufgeteilt in Callbacks welche die Neuerstellung der Batteriedaten benötigen und die sie nicht benötigen, um an Rechenaufwand zu sparen
- Überarbeitung der Struktur, um die Grössenveränderungen möglich zu machen
- Überarbeitung des Dashboards mit CSS

Nach Besprechung

- Erstellung des GitHubs aus dem lokalen Projekt
- Umgestaltung der Code-Struktur für bessere Verständlichkeit
- Abarbeitung der Rückmeldungen aus der Präsentation → siehe nächstes Kapitel

Bearbeitung der Bewertung

Die Nebenkritikpunkte aus der Bewertung waren die folgenden:

- Schlechte Farbwahl für die Titel
- Standardfarben in den Plots
- Tortendiagramm in dieser Anwendung nicht sinnvoll
- Achsenbeschriftungen fehlen noch

Diese wurden alle behoben. Als Hauptkritikpunkte wurden folgende zwei Punkte genannt:


1. Positionierung Visualisierung «Grossverbraucher und Batterie Auslastung»: Die Visualisierung soll bspw. In einen separaten Tab platziert werden, sodass kein Scrolling nötig ist.
2. Gibt es eine intuitive Lösung wie der finanzielle Erfolg für den Betrachter erkennbar ist?

Punkt 1 wurde genauso wie empfohlen umgesetzt. Zudem wurde ein Sawitzki Golay Filter auf die Daten angewendet, um die Spitzen zu glätten und die Daten leserlicher zu machen.

Bei Punkt 2 hatte ich bereits in der Präsentation erwähnt, dass der finanzielle Erfolg nur ein Zwischenresultat ist und daher eine spezielle Signalisierung dessen, wenig Sinn macht. Zudem ist die Zielgruppe des Dashboards Privatpersonen mit passendem Vorwissen definiert. Deshalb macht eine Darstellung der Daten in klaren Zahlen und nicht symbolisch mehr Sinn. Stattdessen wurde die Amortisation, welche das eigentliche Hauptmerkmal des Dashboards ist, zusätzlich noch durch eine Visualisierung erweitert, bei der die «gesparten» Jahre dargestellt ist. Ausserdem wurde die Verteilung der Visualisierung noch so angepasst, dass die Tabelle besser im Zentrum steht.

Anhang



chriddyp  Co-founder of Plotly, Author of Dash

It's not possible to change the rotation of the axis title. However, for units, you should try out `layout.yaxis.ticksuffix` combined with `layout.yaxis.showtickprefix = 'first'` <https://plot.ly/python/reference/#layout-yaxis-ticksuffix> 1.3k