

# Technische Dokumentation – Klimadaten-Dashboard zur Analyse des Vogelzugs in der Schweiz

## 1. Einleitung

Die vorliegende Dokumentation beschreibt die Entwicklung eines interaktiven Web-Dashboards zur Analyse von Vogelzugdaten im Kontext des Klimawandels. Das Projekt wurde im Rahmen der CDK1-Klimadaten-Challenge im Frühlingssemester 2025 umgesetzt und zielt darauf ab, komplexe Zusammenhänge zwischen Klimaszenarien und dem Zugverhalten ausgewählter Vogelarten in der Schweiz verständlich und anschaulich darzustellen.

Unser Team entwickelte ein lokal ausführbares Streamlit-Dashboard, das öffentlich zugängliche Datenquellen, wissenschaftliche Klimaprojektionen und visuelle Darstellungen zusammenführt. Die Lösung richtet sich sowohl an wissenschaftlich Interessierte als auch an Laien mit einem Interesse an Biodiversität und Klimafolgen.

## 2. Projektmanagement und Organisation

Die Zusammenarbeit im Team funktionierte gut. Die Umsetzung erfolgte iterativ in mehreren Phasen. Zu Beginn wurden kleine Tests in Jupyter Notebooks durchgeführt, um sich mit den Daten und Bibliotheken vertraut zu machen. Anschliessend wurden verschiedene Komponenten des Dashboards einzeln in Streamlit erprobt, bevor das finale Dashboard als Ganzes aufgebaut wurde.

Diese schrittweise Vorgehensweise ermöglichte frühes Feedback und gezielte Anpassungen. Für die Versionskontrolle nutzten wir Git, wodurch paralleles Arbeiten und eine saubere Dokumentation des Fortschritts sichergestellt wurden.

## 3. Datenauswahl und wissenschaftliche Grundlage

Für die Umsetzung des Projekts wurden ausschliesslich vertrauenswürdige und qualitativ hochwertige Datenquellen genutzt. Die biologischen Daten stammen aus öffentlich zugänglichen Fachportalen wie [birdlife.ch](https://www.birdlife.ch), [nabu.de](https://www.nabu.de) und [vogelwarte.ch](https://www.vogelwarte.ch). Die Informationen wurden manuell zusammengetragen, bereinigt und systematisch in strukturierte CSV-Dateien überführt. Erfasst wurden zentrale Merkmale der Zugvögel, darunter Zugtyp (z. B. Kurz-, Langstrecken- oder Teilzieher), bevorzugte Temperaturbereiche, typische Ankunfts- und Abflugszeiten, Brutverhalten, Nahrungstypen sowie visuelle Elemente wie Bildpfade und Diagrammpfade. Insgesamt umfasst die finale Datei eine Auswahl von 24 relevanten Arten mit vollständiger Attributbeschreibung.

Ergänzt wurden diese Daten durch wissenschaftlich validierte Klimadaten des **National Centre for Climate Services (NCCS)**. Diese beinhalten modellierte Temperaturprojektionen für verschiedene Regionen der Schweiz bis zum Jahr 2100.

Verfügbar waren drei Klimaszenarien gemäss den internationalen RCP-Konventionen: RCP 2.6 (optimistisch), RCP 4.5 (moderat) und RCP 8.5 (pessimistisch). Die Klimadaten wurden zunächst aggregiert und geglättet, anschliessend in monatliche Durchschnitte überführt und schliesslich mit den artspezifischen Ankunftszeiten verglichen.

Durch diese Kombination konnte das Projektteam eine datenbasierte Hypothese formulieren: Steigende Temperaturen im Frühling könnten bei bestimmten Arten zu einer früheren Rückkehr führen oder – je nach Temperatursensitivität – neue Risiken für Brut und Nahrungssuche schaffen. Das Modell erlaubt damit einen Einblick in mögliche langfristige Auswirkungen des Klimawandels auf das saisonale Zugverhalten heimischer Vogelarten.

#### 4. Datenpipeline und Verarbeitung

Die gesamte Verarbeitung erfolgte in Python mit pandas. Zunächst wurden die CSV-Dateien eingelesen, vereinheitlicht und fehlende Werte ergänzt. Die Temperaturzeitreihen wurden ins Long-Format überführt, sodass sie sich gut mit Monats- oder Art-bezogenen Merkmalen verbinden liessen.

Eine relationale Datenbank wurde nicht verwendet, da die Datenmenge überschaubar und lokal gut verarbeitbar war

#### 5. Technische Architektur und Implementierung

Das entwickelte Dashboard basiert vollständig auf **Python** und wurde mithilfe des Frameworks **Streamlit** realisiert. Streamlit ermöglichte eine schnelle Umsetzung interaktiver Weboberflächen mit minimalem Frontend-Aufwand. Die grafischen Darstellungen und Datenvisualisierungen wurden mit **matplotlib**, **Seaborn** sowie stellenweise **Altair** erzeugt. Letzteres kam insbesondere bei interaktiven oder responsiven Darstellungen zum Einsatz, um z. B. Klimaverläufe oder Komfortbereiche von Vögeln übersichtlich gegenüberzustellen.

Die Applikation ist **modular aufgebaut**: Die Hauptdatei Story.py dient als erzählerische Einstiegskomponente im Scrollytelling-Stil. Sie vermittelt den thematischen Rahmen, stellt Vogelarten vor und führt visuell durch die Klimaprojektionen. Die ergänzenden Dateien Dashboard.py und Temp\_rechner.py bieten klassische Dashboards mit Filterfunktionen, Temperaturberechnungen und interaktiver Visualisierung der Klimadaten.

Die Ordnerstruktur folgt einem klaren Schema mit getrennten Bereichen für:

- **Datenquellen** (Daten/, unterteilt in Bilder, temperatur\_szenarien, Voegeldaten)
- **Seitenmodule** (pages/)
- **Skripte & Hilfsfunktionen** (Story.py, Temp\_rechner.py, test.ipynb etc.)
- **Virtuelle Umgebung** (venv/, lokal ausgeschlossen)
- **Konfigurationsdateien** (.gitignore, requirements.txt)

Ein zentraler Vorteil der gewählten Architektur liegt in der **Plattformunabhängigkeit**. Da keine externe API-Anbindung oder persistente Datenbank notwendig ist, lässt sich das Projekt problemlos lokal ausführen – sowohl auf macOS, Windows als auch Linux. Die Datenhaltung erfolgt rein dateibasiert über strukturierte CSV-Dateien. Zur Installation genügt ein funktionierender Python-Interpreter (Version  $\geq 3.9$ ) sowie das Ausführen von `pip install --user -r requirements.txt`.

Die App wird lokal gestartet via: `streamlit run Story.py`

Durch diesen Ansatz ist das System leichtgewichtig, wartungsarm und eignet sich ideal für explorative Datenprojekte wie dieses.

## 6. Dashboard-Design und Funktionalitäten

Das Layout des Dashboards wurde benutzerfreundlich gestaltet. Über einfache Dropdowns kann eine Vogelart ausgewählt werden. Es folgen relevante Informationen wie Zugzeiten, Komforttemperaturen und die Entwicklung der Temperatur nach Szenario.

Besonders hervorzuheben ist die klare Struktur und das Zusammenspiel von Bildmaterial, Daten und Visualisierungen. Die Anwendung ist funktional, übersichtlich und auch für fachfremde Personen zugänglich.

## 7. Datenstory

Die Story-Komponente (Story.py) bietet einen narrativen Einstieg in das Thema Vogelzug im Klimawandel. Zu Beginn wird allgemein auf den Vogelzug eingegangen, mit einem lokalen Bezug zum Flachsee im Aargau. Darauf folgt die Darstellung des Weissstorchs als Beispielvogel.

Es werden Temperaturentwicklungen unter verschiedenen Klimaszenarien visualisiert, deren Auswirkungen auf den Ankunftsmonat und den Komfortbereich des Storchs analysiert. Zudem wird erläutert, warum Vögel überhaupt ziehen und welche langfristigen Folgen sich durch klimatische Veränderungen ergeben könnten.

Auf interaktive Elemente wurde bewusst verzichtet, um den Fokus auf den Lesefluss und die Narration zu lenken.

## 8. Herausforderungen und Lösungen

Eine der grössten Herausforderungen bestand in der Themenfindung. Die Auswahl geeigneter Datenquellen gestaltete sich anfangs mühsam. Auch die Konzeption einer schlüssigen Storyline benötigte mehrere Anläufe.

Die technische Umsetzung mit Streamlit war neu für das Team, funktionierte nach einer kurzen Einarbeitung aber gut. Das Arbeiten mit neuen Bibliotheken, Unicode-Dateinamen sowie der Daten-Upload auf der FHNW-VM führten zu kleineren

Problemen, die aber alle gelöst werden konnten.

Insgesamt verlief das Projekt erfolgreich. Der Zeitplan war herausfordernd, konnte aber mit guter Kommunikation eingehalten werden. Das Team ist stolz auf das Ergebnis.

## 9. Wissenschaftlichkeit der KI-Ergebnisse

Die im Projekt eingesetzte „Künstliche Intelligenz“ bezog sich auf die unterstützende Nutzung von Sprachmodellen (wie ChatGPT) zur Formulierung von Visualisierungstexten, Interface-Beschriftungen sowie zur inhaltlichen Strukturierung der Story-Komponenten. Die zugrunde liegenden Berechnungen und Diagramme basieren jedoch ausschliesslich auf nachvollziehbaren Datenverarbeitungsprozessen in Python und beinhalten keine Black-Box-Algorithmen oder maschinelles Lernen zur Vorhersage.

Die Ergebnisse lassen sich vollständig durch die verwendeten Datensätze, mathematischen Operationen und manuell definierten Bedingungen (z. B. Komforttemperaturbereiche der Vögel) nachvollziehen und validieren. Damit erfüllt das Projekt grundlegende Kriterien wissenschaftlicher Nachvollziehbarkeit.

## 10. Fazit und Ausblick

Das Projekt demonstriert eindrücklich, wie **wissenschaftlich fundierte Klimadaten** mit **biologischen Beobachtungsdaten** verknüpft und in einer zugänglichen, visuellen Form aufbereitet werden können. Die Verbindung von **regionalen Klimaszenarien** mit **artenspezifischen Ankunftszeiten und Temperaturpräferenzen** erlaubt neue Einblicke in die potenziellen Auswirkungen des Klimawandels auf das Zugverhalten von Vögeln in der Schweiz.

Besonders hervorzuheben ist der **zweigeteilte Aufbau der Anwendung**: Die narrative Einführung im Scrollytelling-Stil (Story.py) schafft einen emotionalen, gut verständlichen Zugang zum Thema – insbesondere auch für Laien oder interessierte Beobachter:innen. Ergänzt wird dies durch ein klassisches, interaktives Dashboard, das detaillierte Analysen ermöglicht und Filterfunktionen für spezifische Fragestellungen bietet. So wird sowohl ein **niedrigschwelliger Einstieg** als auch eine **tiefergehende Datenexploration** ermöglicht.

Durch die **klare technische Architektur**, die **lokale Ausführbarkeit** und die **transparente Datenaufbereitung** ist das Projekt gut dokumentiert, reproduzierbar und bietet zahlreiche Anknüpfungspunkte für Weiterentwicklungen. Ein möglicher nächster Schritt wäre die Integration weiterer Tiergruppen (z. B. Insekten, Amphibien), die Einbindung einer Datenbank zur flexibleren Abfrage oder ein öffentliches Deployment via Streamlit Cloud oder Render.

Das Projektteam zieht ein durchwegs positives Fazit: Neben fachlichen Erkenntnissen wurde auch der praktische Umgang mit Daten, Visualisierungsframeworks und wissenschaftlicher Kommunikation geschult.

