Folie 1 – Analyse des Datensatzes

Analyse des Datensatzes: Bewertung der Authentizität & Implikationen

Untertitel: Eine forensische Datenanalyse für den IT-Vorstand

Folie 2 – Python Module

mod\_010\_laden\_reinigen.py

mod\_020\_csv\_analyzer.py

mod\_040\_feature\_engeneering.py

mod\_041\_Umrechnung\_wert\_ppm\_µgm3.py

mod\_042\_glaetten\_der\_sensordaten3.py

mod\_050\_datenanalyse plotten.py

mod\_051\_karte\_luft\_qualitaet.py

mod\_052\_zeitslider.py

mod\_053\_korrel\_und\_top\_10.py

mod\_080\_text\_generieren.py

mod\_100\_gui.py

gps2street.py

Authentizitäts-Score zu Beginn: M-Score: [m] | R-Score: [r]

Folie 3 – Datenpipeline Schritte

Projekt- und Datenordner bestimmen

– Ordner „bearbeitet“ relativ zum Projektspeicherort suchen

Erste CSV-Datei finden

– Erste .csv-Datei im Ordner suchen

Headerzeile suchen

– Datei zeilenweise einlesen, Header identifizieren, # überspringen

Datenzeilen bereinigen

– „;“ durch „,“ ersetzen

– „ MESZ“ und „ UTC“ entfernen

Daten in DataFrame laden

– Bereinigte Zeilen direkt in pandas DataFrame laden

Datumsspalten konvertieren

– DateTime und GPS\_DateTime zu Datetime-Objekten

Erste X Minuten filtern

– Zeilen innerhalb der ersten X Minuten filtern

Zeilen ohne GPS-Daten entfernen

– Leere oder ungültige GPS\_Lat/GPS\_Lon/GPS\_Alt löschen

GPS\_Course-Korrektur

– Werte >3-stellig auf 0 setzen

Zeilen mit GPS\_Lon < 8 entfernen

Letzte Zeile entfernen (fehlerhafte Messung)

Zielordner & Dateinamen bestimmen

– Ordner „bearbeitet0“ + neuer Dateiname mit Zeitstempel

DataFrame als CSV speichern

Anzeigeoptionen setzen (pandas)

DataFrame zurückgeben

Optional:

– Spaltennamen & erste Zeilen anzeigen

– Transponierte Statistik

– Histogramme erstellen

– „Fertig!“ ausgeben

Folie 4 – AirScout Environmental Monitoring System

Mobiles, hyperlokales Umweltmonitoring (Arduino + Python)

Messung: Luftqualität, Radioaktivität, Lärm, Mikroklima & GPS

Datenlogging auf SD-Karte → CSV → Kaggle-Dataset

Python-Pipeline für Analyse, Heatmaps & Kartenvisualisierung

Einsatz für Forschung & Citizen Science

Folie 5 – AirScout Hardware & Firmware

Arduino Mega 2560 mit 12+ Sensoren

MQ-Gassensoren (CO/CO2/NOx/LPG/etc.)

GPS (200Hz), Geigerzähler & Dual-Mikrofone

Temperatur, Luftfeuchte & Lichtmessung

SD-Karten-Logging mit RTC-Zeitstempel

Firmware: Multisensor-Sampling & Plug&Play-Kalibrierung

Folie 6 – Beweis B: Zeitmuster

Analyse der Zeitstempel: Fast alle Transaktionen zwischen 08–20 Uhr CST

Lücke zwischen 22–03 Uhr – kein globaler Fraud folgt diesem Muster

Hinweis auf künstliche Generierung durch ein Skript

Folie 7 – AirScout Python Datenpipeline

CSV-Preprocessing & Datenbereinigung

Analyse in Jupyter Notebooks

Plotly-Heatmaps & Geo-Visualisierungen

Export als Kaggle-Dataset & GeoJSON

Schneller Start via pipeline.py & requirements.txt

Folie 8 – AirScout Nächste Schritte & Citizen Science

Dokumentation der MQ-Sensor-Kalibrierung

Veröffentlichung des Kaggle-Datasets (CC-BY 4.0)

Tutorial für Bürgerwissenschaftler

Einsatz in Forschung, Umweltprojekten & Lehre

Quelle: AirScout Project (2025)

Folie 9 – AirScout-Analytics Umweltkontrollsystem

Sensordatenanalyse & Dashboard für Neustadt/Wstr.

Messung: Temperatur, Gase, Licht, Lärm & Strahlung

GPS-gestützte Zeitreihen- & Kartendarstellung

Interaktive Visualisierungen & Filterfunktionen

Kaggle-Dataset: Hyperlocal Urban Environmental Monitoring

Folie 10 – AirScout-Analytics Tech Stack

Python 3.x, Pandas & Numpy für Datenverarbeitung

Matplotlib & Seaborn für Standardplots

Plotly/Dash oder Streamlit für interaktive Dashboards

GeoPandas & Folium für Kartendarstellungen

Jupyter für Analyse & Git/GitHub für Versionierung

Folie 11 – AirScout-Analytics Kernfunktionen

Automatisierte Datenpipeline für CSV-Sensordaten

csv\_analyser.py: Analyse + Visuals (HTML/PDF)

geo\_mapper.py: Interaktive Pollution-Maps (GeoJSON)

report\_generator.py: Automatische Reports (PDF)

Erweiterbares Dashboard für Echtzeit- & Offline-Analysen

Folie 12 – AirScout-Analytics Prüfung & QS

MVC-Architektur & Unit-Tests (pytest)

CI/CD mit GitHub Actions

Zeitreihenanalyse, Geospatiale Visualisierung & Signalverarbeitung

Testabdeckung aktuell: 0% (Geo-Mapping & Reports)

MIT-Lizenz & Projekt für Python-Prüfung 2025

Folie 13 – AirScout-Analytics Umweltkontrollsystem

Umweltmessungen in Neustadt/Wstr.

Sensorik für Temperatur & Luftfeuchtigkeit

Gase: CO, CO2, Methan & Luftqualität

Erfassung von Licht & Lärmbelastung

Strahlung in CPS (Radiation Monitoring)

Folie 14 – Datenmanagement & Analyse

Erfassung via Arduino & SD-Karte

Übertragung & Speicherung als CSV

Analyse in Python & Power BI

Automatische Bereinigung & Transformation

Langzeitarchiv & Trendanalyse

Folie 15 – GPS-gestützte Auswertung

Jede Messung erhält Geo-Koordinaten

Routenaufzeichnung durch die Stadt

Zeitreihenanalyse auf Karten

Erkennung von Hotspots & Anomalien

Exportierbare Heatmaps & Animationen

Folie 16 – Interaktive Visualisierungen

Dashboards mit Filter- & Drilldown

Zeitreihen für Temperatur & Gase

Livekarten mit Messpunkten

Kombination aus Diagrammen & Karten

Export als PDF oder Präsentation

Folie 17 – Veröffentlichung & Nutzung

Kaggle-Dataset: Hyperlocal Urban Environmental Monitoring

Open-Data für Forschung & Städteplanung

Vergleich mit Wetter- & Umweltstationen

Einsatz für Bürger- & Schulprojekte

Basis für smarte Stadtentwicklung

Folie 18 – Anhang A2 Reporting in Power BI

Übersicht ? Betrugs-Detail ? Kunden-Profil

Filter, Drillthrough, Tabs für Betrug/Kunde/Trends