Gero Krikawa| Erich-Kästner-Straße 11 | 68723 Oftersheim

Projektauszüge Data Science / Data Engineering
Python, SQL, SQL-Server, AWS-Cloud, Spark, SAP-HANA

Inhalt

Verwendete Python Bibliotheken: pandas, numpy, sklearn, tensorflow, plotly express, matplotlib, pyspark, hana-ml

GitHub Link: Projektauszüge inkl. Coding und Dokumentation

1 Python / Data Engineer / Data Quality

Praxis Projekt: Visualisierung von Klimadaten auf einer Benutzeroberfläche
Teilprojekt: Data Engineering (Datenquelle, Datenintegration, Datenmodellierung, Datenqualität):
Ziel: Rohdaten mit diversen Data Quality Problemen extrahieren, bereinigen und in geeigneter Zielstruktur ablegen
Entwicklungsumgebung: Pycharm, Visual Studio Code
Programmpiersprache: Python (pandas, matplotlib, pandas_profiling, meteostat, plotly express, json…)

Vorgehensweise:

- Geeignete Rohdatenquellen suchen und Schnittstelle implementieren
- Data Quality: Rohdaten auf Data Quality prüfen (Formate, Vollständigkeit, Ausreißer…) und Konzeption von Maßnahmen zur Sicherstellung der Datenqualität
- Datenintegrationsprozess: Erstellung von eigenen Funktionen zur Datenbereinigung
- Datenmodellierung Zielstruktur festlegen und implementieren
- Bereitstellung einer sauberen Daten Basis
- Entwurf einer Benutzeroberfläche zur Visualisierung der Klimadaten
- Erfahrung sammeln mit Big Data
- Test, Dokumentation und Präsentation des Ergebnisses

GitHub Link: Projektauszüge inkl. Coding und Dokumentation

2 Data Engineer / SQL / Datenmodellierung / Data Warehouse / Data Vault / Data Quality

Praxis Projekt: Digitalisierung eines bisher analog geführten Unternehmens auf Basis schriftlicher Dokumente und Befragung des Kunden.

Konzeption, Entwicklung eines Operatives Systems (Datenmodell) sowie geeigneten Data Warehouse Modells inkl. ETL Prozessen und umfangreicher Maßnahmen zur Sicherstellung der Datenqualität

Datenbank: SQLite
Programmiersprache: SQL

Vorgehensweise

- Business Understanding / Data Understanding
  Sehr Umfangreiche Anforderungsanalyse:
  - Analyse und Bewertung von gegebenen schriftlichen Teilanforderungen
  - Umfangreiche Interviews mit dem Kunden zur Klärung offener Fragen zur Klärung der Anforderung und Erstellung eines Lastenheftes zur Abnahme vom Kunden
  - Abnahme des Lastenheftes durch den Kunden

- Data Understanding: Definition der notwendigen Entitäten
- Datenmodell operatives System:
  Auf Basis von schriftlich und mündlich ermittelten Anforderungen und dem daraus erstellten Lastenheft: Konzeption und Entwicklung (Programmierung) eines geeigneten Datenmodells (Entity Relationship Modell)
- Datenmodell Data Warehouse:
  Entwurf eines auf Basis der Anforderungen passenden Data Warehouse Datenmodells (Data Vault)
- Entwurf von Data Pipelines zur Befüllung der Datenmodelle
- Entwurf eines Data Quality Konzeptes basierend auf den individuellen Gegebenheiten inklusive Vorschlägen zu Maßnahmen der Sicherstellung der Datenqualität
- Präsentation der Ergebnisse

  GitHub Link: Projektauszüge inkl. Coding und Dokumentation

3 Data Analyst / Python / Explorative Datenanalyse / Data Engineer / Data Quality

Praxis Projekt: Durchführung einer explorativen Datenanalyse inkl. Data Engineering
Datensatz: Kiva
Entwicklungsumgebung Jupyter Notebook
Programmiersprache: Python Bibliotheken (numpy, pandas, plottly express)

Vorgehensweise:

Verwendung des Modells: CRISP DM (Cross Industry Standard Process for Data Mining)

- Business Unterstanding
- Data Understanding
- Data Preparation (Data Quality)
- Modelling
- Evaluation
- Deployment

Teil 1: Data Engineering

- Erkennung und Bereinigung von Dubletten inkl. Begründung der Vorgehensweise
- Erkennung und bei Bedarf Bereinigung von fehlenden Werten inkl. Begründung der Vorgehensweise
- Ermittlung von Ausreißern inkl. Beurteilung und Maßnahmen
- Ergänzung der Rohdaten durch weitere exteren Daten
- Ermittlung von berechneten Kennzahlen

- Erstellung eines Dataframe für die Explorative Datenanalyse (EDA)

Teil 2: Data Analyst

Explorative Datenanalyse auf Basis von Business Fragestellungen

- Daten selektieren, gruppieren, aggregieren
- Erstellung von Plots & Findung von Mustern
- Deskriptive Analyse
- Analytische Insights
- Entwurf eines Dashboards
- Zusammenfassung der gewonnenen Insights und Präsentation der Ergebnisse

GitHub Link: Projektauszüge inkl. Coding und Dokumentation

4 Machine Learning / Python / Klassifikation/ knn, Logistische Regression, Support Vector Machine, Random Forest

Praxis Projekt: Auf Basis eines gegebenen Datensatzes: Healthcare Dataset Stroke Evaluation und Optimierung verschiedener ML-Algorithmen zur Vorhersage eines Schlaganfalls und Bewertung anhang geeigneter Fehlermetriken. Welcher ML-Algorithmus ist unter welchen Rahmenbedingungen am geeignesten in der praktischen Anwendung?

Datensatz: Healthcare Dataset Stroke
Entwicklungsumgebung: Pycharm
Programmiersprache: Python Bibliotheken: pandas, numpy, matplotlib, sklearn, imblearn

Vorgehensweise:

- Business Understanding
- Data Understanding
  - Strukturelle, Inhaltliche Anpassung, Splitting
  - Oversampling mit SMOTE
  - Definition geeigneter Fehlerkoeffizienten und Begründung

Auswahl von ML-Algorithmen und Entwicklung eines Modells zur Evaluation jedes Algorithmus

1. Knn
2. Logistische Regression
3. Support Vector Machine
4. Random Forest

- Vergleich und Bewertung der Algorithmen
- Voting Verfahren
- Lessons Learned
- Dokumentaiton und Präsentation

GitHub Link: Projektauszüge inkl. Coding und Dokumentation

5 Deep Learning / Python / Predictive Modeling - RNN/LSTM

Praxis Projekt: Forecasting / Zeitreihenvorhersage Neural Network: RNN/LSTM
Datensatz: Univariate Zeitreihe, saisonal
Entwicklungsumgebung Jupyter Notebook
Programmiersprache: Python (numpy, pandas, matplotlib, sklearn, tensorflow, json, hana_ml, pottly express..)

Vorgehensweise

- Business Unterstanding
- Data Understanding
- Google 7 steps of Machine Learning in practice (TensorFlow 2.0)
  - Gathering data
  - Preparation data
  - Choosing a model
  - Training
  - Evaluation
  - Hyperparameter Tuning
  - Prediction
- Beurteilung der Ergebnisse
- Dokumentation und Präsentation

GitHub Link: Projektauszüge inkl. Coding und Dokumentation

6 Data Engineer / SAP HANA Cloud Machine Learning mit Python

Praxis Projekt: Machine Learning mit SAP HANA und Python
Datensatz: Index Industrial Production: Manufacturing: Non-Durable Goods: Ice Cream and Frozen Desserts
Entwicklungsumgebung Jupyter Notebook
Programmiersprache: Python (numpy, pandas, matplotlib, sklearn, tensorflow, json, hana_ml, pottly express..)

Vorgehensweise

- Data Engineering: SAP ←→Python
- Anbindung Quellsystem SAP HANA Cloud an Python (HANA_ML)
- Entwurf Datenintegrationsprozess für Hana Cloud Tabelle
- Entwurf Datenmodell in Python zur Ablage der Tabelle
  (SAP HANA Dataframe -> PANDAS Dataframe)
- Überblick SAP HANA Machine Learning Möglichkeiten und Beispiel Anwendung
- Dokumentation und Präsentation

GitHub Link: Projektauszüge inkl. Coding und Dokumentation

7 Data Engineer / SQL /SQL Server/ Datenmodellierung / ER-Modell / Stored Procedures / DB Trigger

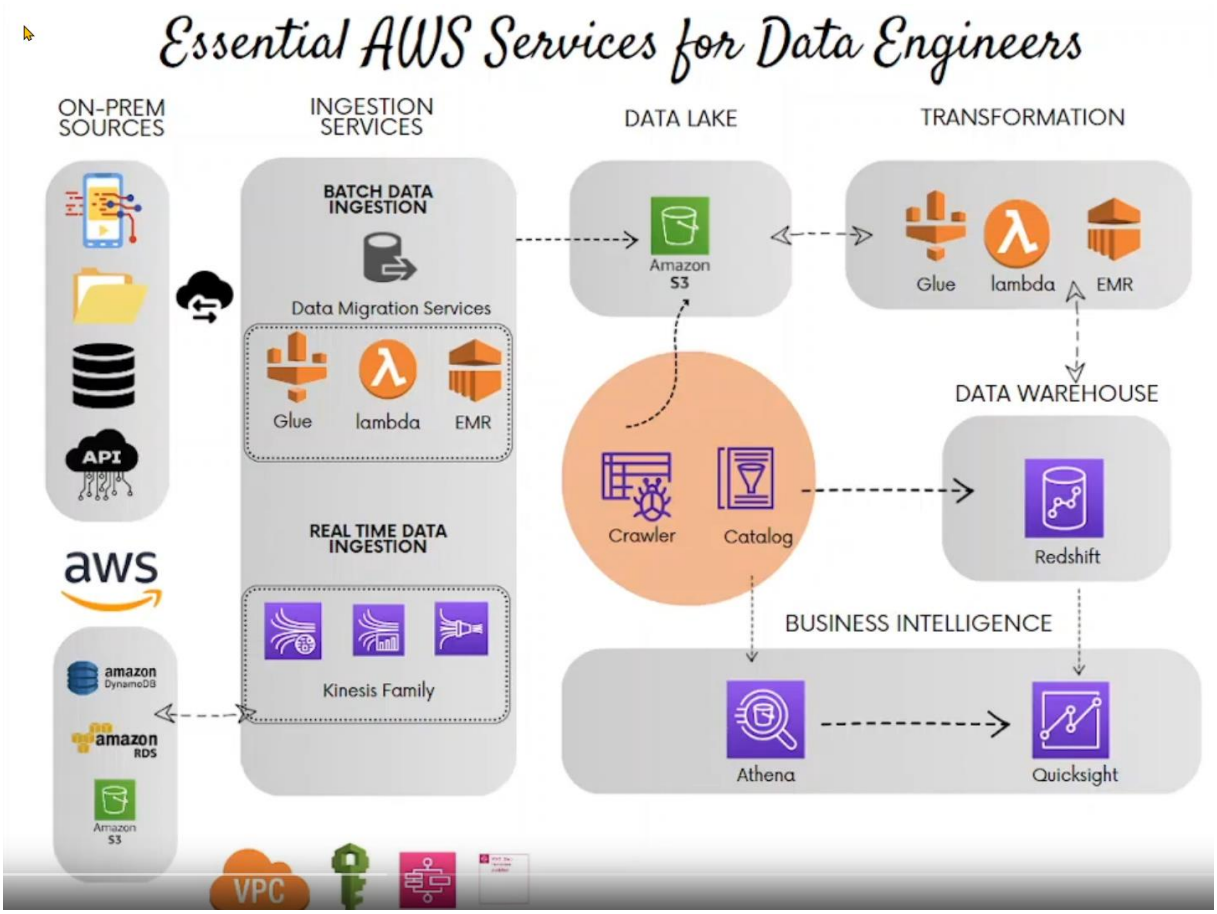
Praxis Projekt: Entwurf eines Buchungssystem für Hotel und Restaurant
Datenbank: MS SQL Server 2019, MS SQL Server Management Studio
Programmiersprache: SQL

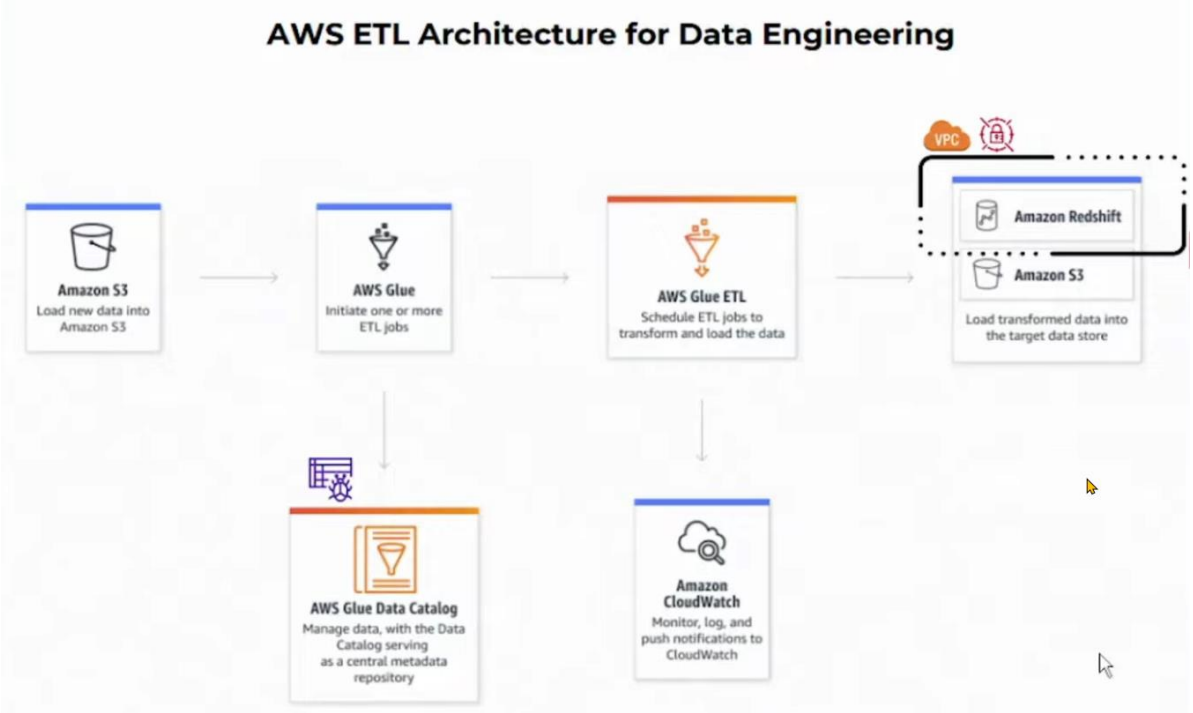
Vorgehensweise:

- Business Understanding: Definition von Business Scenarios und Ableitung des Prozesses
- Data Understanding: Definition der notwendigen Tabellen und Beziehungen
- Entwurf eines relationalen Datenbank Modells (Entity Relationship Modell)
- Befüllung des Datenbank Modells mit geeigneten Daten
- Konzeption der Logik und Entwicklung von: Stored Procedures, Funktionen, Datenbank Trigger…
- Test des Zusammenspiels der Logik im Buchungssystem
- Dokumentation und Präsentation des Ergebnisses

GitHub Link: Projektauszüge inkl. Coding und Dokumentation

8 Data Engineer – AWS Cloud - end to end project - AWS Cloud, Datapipelines, Apache Spark, Pyspark, Jupyter Notebook, Redshift





Project description
Data-Engineer end to end project using AWS Cloud.
Using of infrastructure as a Code for creation of necessary AWS infrastructure.
Uploading raw data into S3 Data Lake. Building an end to end Data Pipeline using Pyspark for loading Data from S3 Source, Transformation, Aggregation, Data Quality, writing in AWS Redshift. Data Warehouse

Used technologies:

- AWS Cloud
- Infrastructure as a Code to build the necessary AWS infrastructure
- AWS S3 Data Lake
- AWS Glue
- Apache Spark
- Python Pyspark
- Data Pipeline (Interactive Jupyter Notebook / Pyspark)
- AWS Redshift Data Warehouse
- IAM Authorization

Project Workflow

- Create Data Engineering System in AWS using „Infrastructure as Code“
- Test the created infrastructure
- Develop the necessary components (crawler, database..)

- Create an End to End Data Pipeline
  - Upload raw data (Source File: sales_records.csv) into S3 Data Lake (Data Source)
  - Destination: AWS Redshift Data Warehouse
  - S3 Data Lake Storage & Source of Data Pipeline
  - AWS Clue – using crawler to catalog data
  - Processing data using Pyspark within interactive Jupyter Notebook in Glue

- Build Data Pipeline in Pyspark using Glue Jupyter interactive Notebook (Data Quality, Transformation, Aggregation…)
- Reading data from s3 Data Lake storage, processing it via Data Pipeline in Spark and then loading it into AWS Redshift Data Warehouse (using dynamic frames and spark data frames

GitHub Link: Projektauszüge inkl. Coding und Dokumentation