Building end to end Analytical tool to analyze trend in AnaCredit Granular Regulatory Reports which includes:

* Python Scripts to extract data from AnaCredit reports in xml format and forming corresponding delimited files
* Stored Procedure / Scripts in MS SQL Server to:
  + Load delimited files to non-live tables using BULK INSERT and METADATA files
  + Validate data loaded to non-live tables
  + Copy data from non-live tables to live tables
* Views on underlying tables to analyze trend in data in AnaCredit report of past 13 months
* PowerBI dashboards to visualize the output of views built
* Unix FileWatcher script to poll for AnaCredit reports in processing directory and start the process

Building required software in an ETL and Reporting Tool called Axiom Controller View (Axiom CV) to generate Granular Regulatory Reports of AnaCredit, Residential Real Estate (RRE) and Commercial Real Estate (CRE) which includes building or implementing:

* Business Rules published by Regulatory Bank (De Nederlandsche Bank) using different Axiom CV objects
* Regression suites to find deviations in final reports of previous month (or quarter) to current month (or quarter)
* Building validation modules to validate each of the reports generated by Axiom CV
* Building workflow to encrypt zip file containing final reports into a mime file

Implementing CICD pipeline in Axiom CV tool using its ReST API which includes automating:

* Creating development branches automatically on a monthly basis.
* Periodically (per day) commit the finalized components into centralized release branch by automatically exporting the components from development branch and importing them to centralized release branch.
* Compare the imported objects in release branch with its counterpart in development branch from which it was imported to check if the object was correctly committed to release branch.

Notify the developers through an automated e-mail in case of incorrect commit.

* Automatically exporting the objects from pre-production environment and importing it to production environment

Personal Learning Projects:

[New York City Taxi Dataset Analysis in PySpark](https://github.com/K-Suhas-Bharadwaj/TaxiDataset/blob/main/nyc_taxi_trip_analysis.ipynb)

By analyzing the New York City Taxi Dataset that was released by Google on public platform, answers to the following questions were drawn using PySpark Data Structures such as RDD and DataFrames.

* How have the number of trips changed over time?
* How does taxi usage differ by day and by time of the day?
* What are the top trip destinations and where are those trips coming from?
* A model was trained to predict the fare of a given trip using Spark Mllib.

Implementing Steganography using different techniques

Steganography is a process of concealing the secret information into cover object such as text,

audio, image, and video. This process has been achieved in two ways:

* [Using an AI Meta Heuristic Search Algorithm called Cuckoo Search.](https://github.com/K-Suhas-Bharadwaj/Steganography_Cuckoo_Search/blob/main/Cuckoo.py)

This technique mimics the intelligent behavior of Cuckoo birds which show their intelligence in raising their offspring in the nest of other host birds.

We use hamming distance between pixels as heuristic to guide our random search of detecting best pixels of cover image to hide our secret data.

* [Using Convolutional Neural Networks](https://github.com/K-Suhas-Bharadwaj/Deep-Steganography/blob/main/deepsteganography.py)

In this technique the convolutional neural network is trained to identify different patterns in cover image and secret image and embed the almost matching pattern of secret image in the corresponding pattern of cover image.

This helps in maintaining the quality of cover image.

Solving Sudoku puzzle using randomized search algorithms

Solving sudoku puzzle using backtracking algorithm takes exponential time. This can be reduced by solving it using randomized search algorithms.

Sudoku puzzle is solved using:

* [Simulated Annealing Algorithm](https://github.com/K-Suhas-Bharadwaj/Sudoku-using-AI-algorithms/tree/master/Sudoku/simulatedAnnealing)
* [Genetic Algorithm](https://github.com/K-Suhas-Bharadwaj/Sudoku-using-AI-algorithms/tree/master/Sudoku/geneticAlgorithm)