20 November 2020

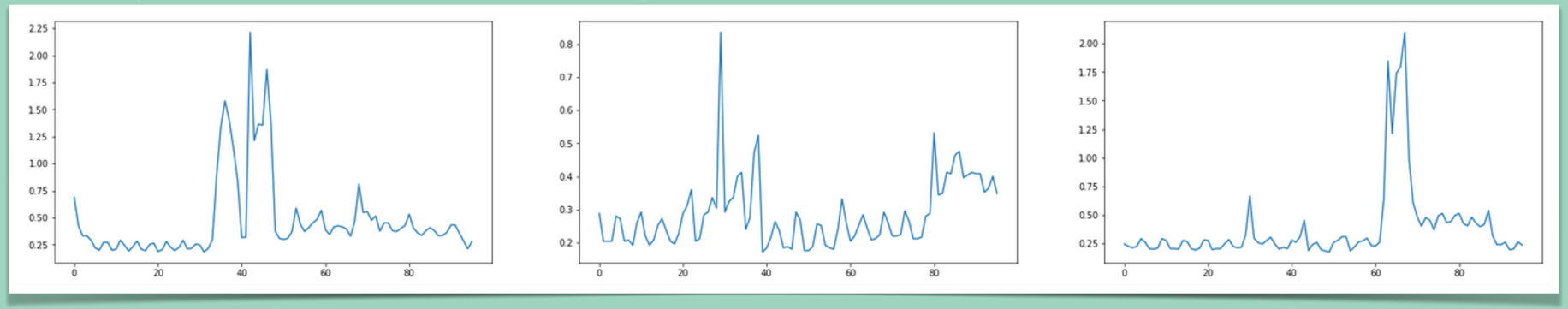
Creos Challenge Report

by KONCHENKO Stanislav

Table of content

1. Short description of the challenge

A smart meter reports the electricity consumption or production of a customer in 15-minute intervals. With 24 hours in a day, this results in 24*4 = 96 values per customer per day. The value is not cumulative, the value at a give time represents the consumption/production during a 15-minute interval. This series of values in a given timespan (e.g. a day) is called a load profile or load curve, for example these:



The load profile a household generally looks different than that of a commercial restaurant. Before smart meters existed, a set of standard load profiles was conceived to approximate the load profile of different types of customers. With smart meters, an interesting question could be whether new patterns can be detected in the real load profiles.

The challenge

In this challenge, there is a (fake) dataset in the csv format. The dataset contains 1000 samples of load profiles, each load profile comprises 96 values. Each profile was created by taking one of three *base* profiles and adding random noise to it. The task is to analyse whether you can identify clusters in the dataset by using the good practices one'd expect from a data science project, by documenting methods and making experiments repeatable.

2. Cluster analysis.

A machine learning project may not be linear, but it has a number of well known steps.

ML project stages

1. Define Problem.
2. Prepare Data.
3. Evaluate Algorithms.
4. Improve Results.
5. Present Results.

2. Cluster analysis. Define problem

Defined Problems and taken Decisions

✓ Clustering is an unsupervised problem of finding natural groups in the feature space of input data.

✓ There are many different clustering algorithms and no single best method for all datasets.

✓ Input data - set of time series. This is a constraint when you start to use well-known libraries for clustering (e.g. scikit-learn Python library)

✓ By default, python is a language for data analysis (wide amount of Data Science, Machine Learning, Deep Learning, data manipulation libraries, easy setup and maintenance).

✓ To work with set of time series I chose tslearn - machine learning toolkit for time series analysis in Python. Has 1.4 K stars on GitHub, implemented clustering algorithms and set of utils to manipulate raw data. Manipulates with arrays (instead of pandas data frames)

✓ I decided to prepare a python cmd program which can operate with different input files, user can choose which algorithms and preprocessing steps to use for clustering. This program saves results as well as visual analytics in output folder. Its setup is very convenient by using pipenv (new python package to manipulate with virtual environments, modules, dependencies, etc)

2. Cluster analysis. Prepare data

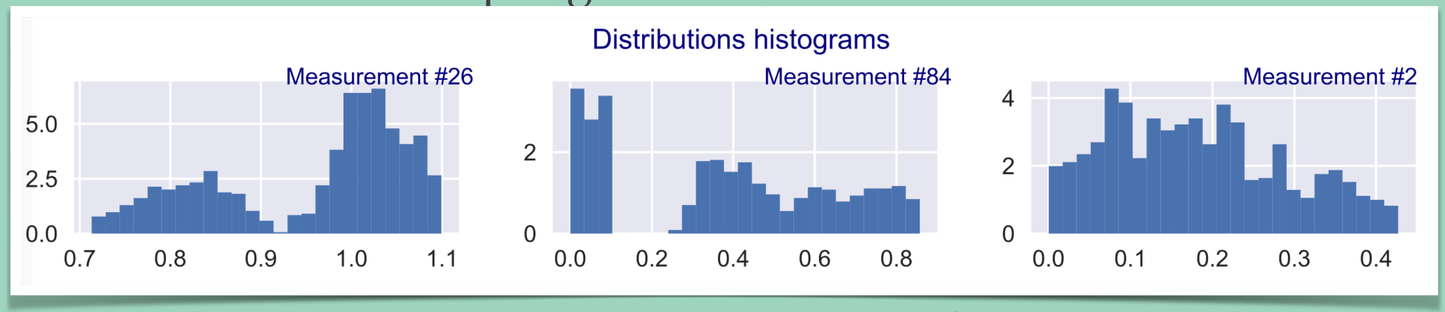
Preprocessing data:

✓ First step - load input data - I use **pandas.*read_csv***.

✓Second step - convert loaded data to the format needed to operate with chosen toolset. In our case I can use utility function from the **tslearn.utilities.*to_time_series_dataset*** which has one important feature - it adjusts the size of all time series to the largest one by adding NaN values

✓ Some algorithms do not manipulate well with NaN. To substitute NaNs I interpolate time series and predict new values

✓ Very important question - normalise (Min-Max scaling) or standardise (mean removal and variance scaling) input data. To answer it I draw 3 distribution histograms of randomly chosen samples which can help to answer which method of the sampling is needed to use.



✓ In our case looks like the distribution are close to the normal / Gaussian. So best way to use standardisation.

✓REMARK: better to try all type of sampling, as well as run program without it to see better results.

2. Cluster analysis. Evaluate algorithms

Used algorithms:

✓ Well known and wide used clustering method is K-means. K-means is a common clustering algorithm that constructs clusters of data by splitting samples into k groups and minimising the sum-of-squares in each cluster

✓ The distance measures used in standard clustering algorithms, such as Euclidean distance, are often not appropriate to time series. A better approach is to replace the default distance measure with a metric for comparing time series, such as Dynamic Time Warping.

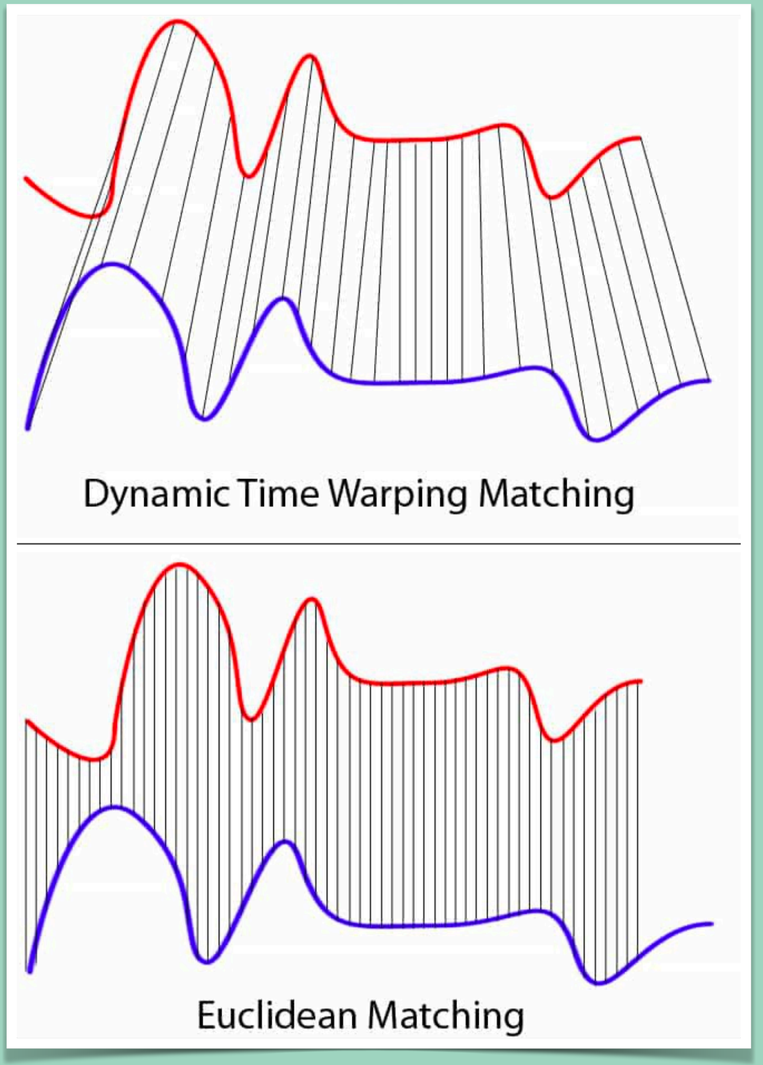
✓ DTW is calculated as the squared root of the sum of squared distances between each element in X and *its nearest point in Y*. Complexity O(MN)

✓ soft-DTW is a differentiable variant of DTW that replaces the non-differentiable ***min*** operation with a differentiable ***soft-min*** operation

✓ In our case it allows to make smoother the resulting metric

✓ Another metric which I used for clustering is Global Alignment Kernel metric.

✓ Other implemented algorithms is k-shape clustering.



2. Cluster analysis. Improve results

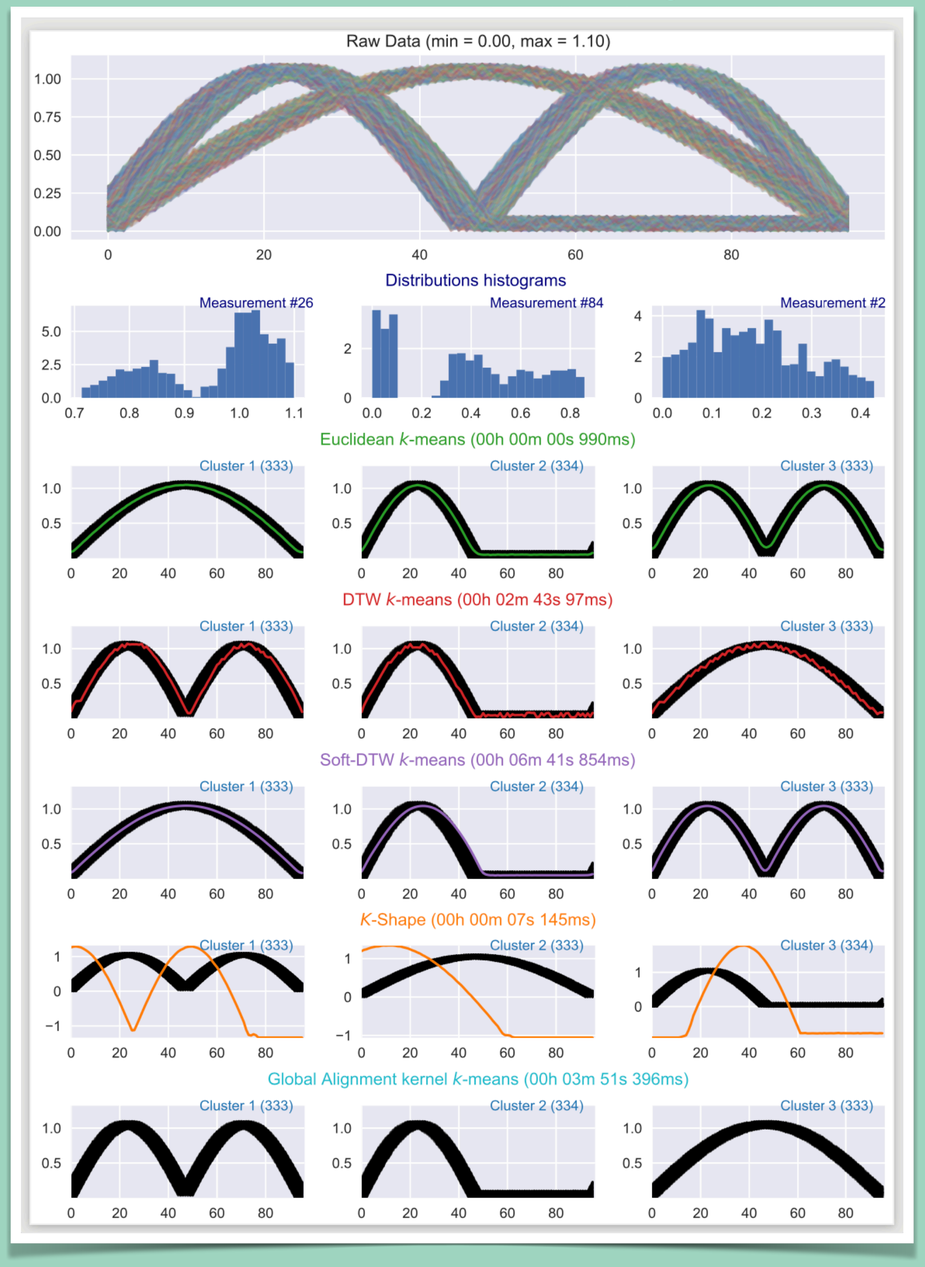
✓ Proposed python script allows to use different preprocessing steps before performing the clustering.

✓You can easily change arguments of the models in the code.

✓ Also it is possible to implement hyper-parameter tuning of the models in pipeline (future improvement)

✓ Since the results of the clustering are equal to 1.0 there is no needed improvement.

2. Cluster analysis. Present results



- ✓ Attached pdf represents result of the executing the clustering solution.
- ✓ All algorithms show 100% accuracy
- ✓ The resulted csv with classes also save in out folder. I used them to fit/train for classification models.

3. Classification. Define problem

✓ How can someone use the obtained results from previous stage?

✓ Need a solution of the practical using of the clustering

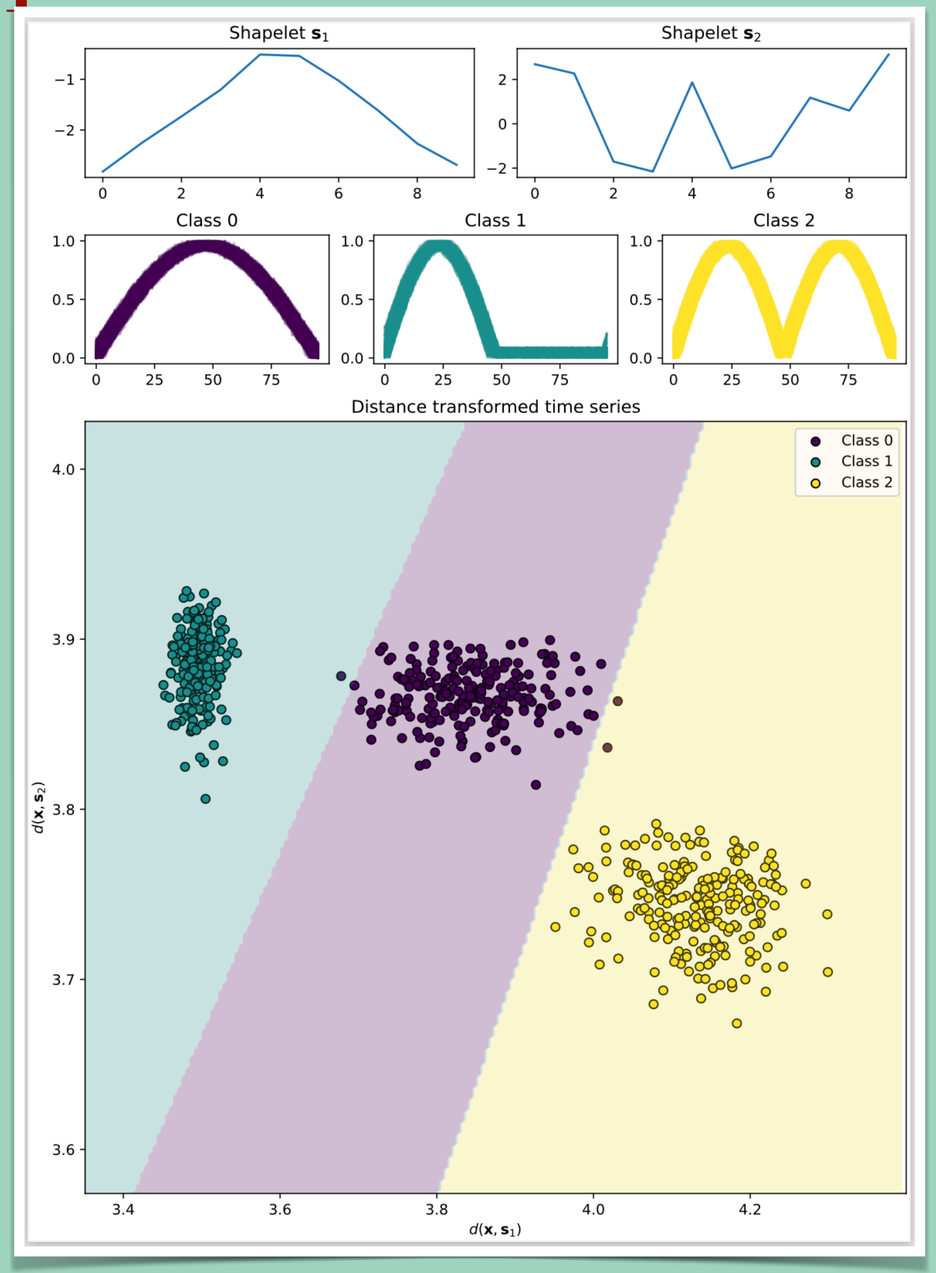
✓ I decided to extend a challenge and develop a classification platform based on the result of clusterization

Main components

1. Splitting script - randomly split input and clustering results into train and test datasets in proportion 67:33
2. Trainers - scripts with 4 implemented classification algorithms:
   - K Nearest Neighbours TimeSeries Classifier (most known and widely used)
   - TimeSeries Support Vector Classification model that uses GAK as kernel
   - Learning Shapelets Time Series Classifier
   - Multi-Layer Perceptron neural network classifier for time series
3. REST API application for classification (Python Flask):
   - input parameters: type of the model and time series
   - response: JSON with class
4. Simple console client to test REST API application. Used test data and compare results from API with tested.

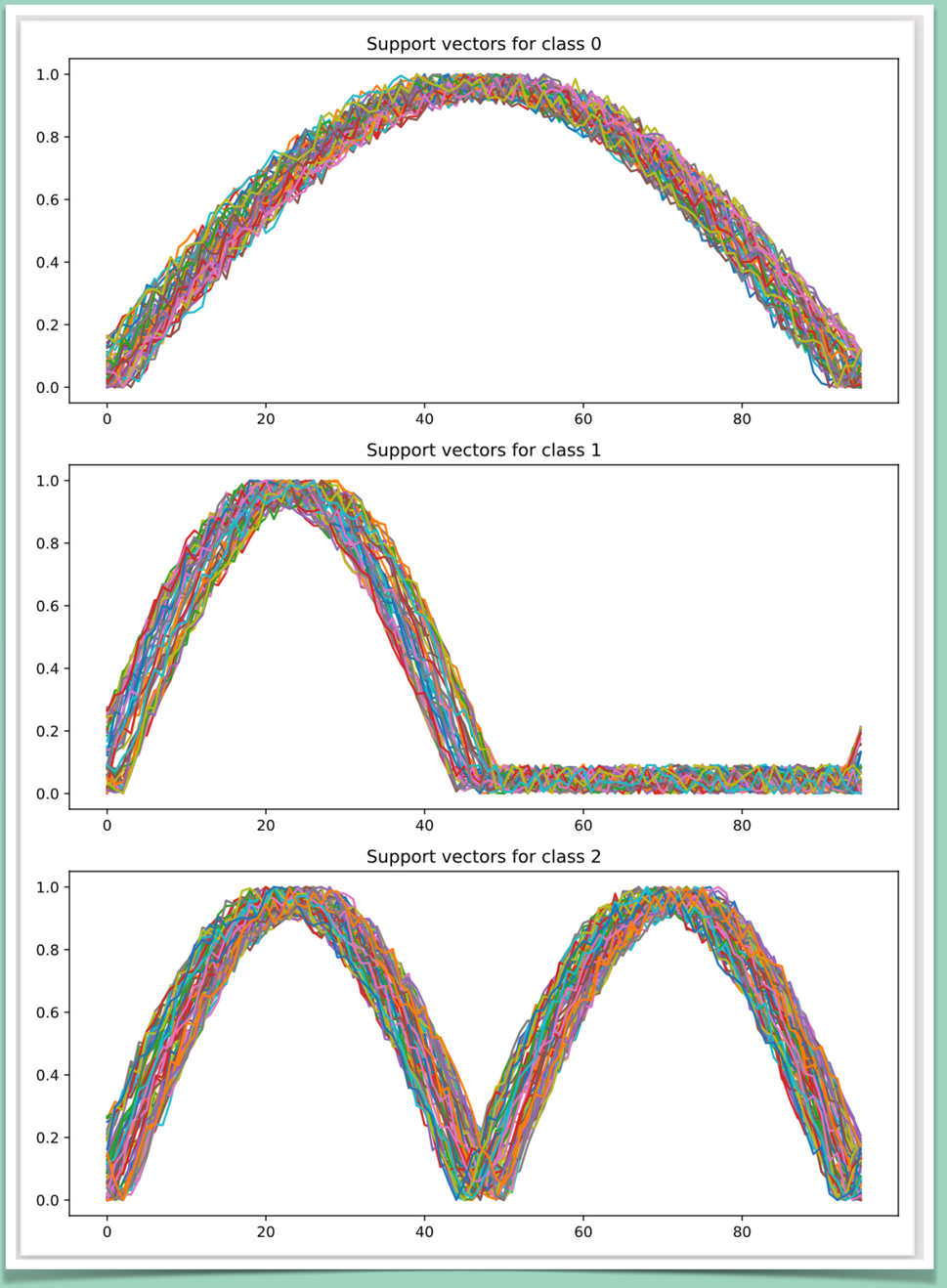
3. Classification. Learning Shapelets

- ✓ Learning Shapelets method used in order to learn a collection of shapelets that linearly separates the timeseries.
- ✓ I extract two shapelets which are then used to transform our input time series in a two-dimensional space, which is called the shapelet-transform space
- ✓ I plot the decision boundaries of this classifier for each of the different classes.
- ✓Could be interested with real data



3. Classification. GAK SVM

- ✓ a *TimeSeries Support Vector Classification* model that uses Global Alignment Kernel as kernel is fit a
- ✓the support vectors for each class are reported.



3. Classification. Multi-layer Perceptron neuro network classifier

✓ This model optimises the log-loss function using LBFGS or stochastic gradient descent.

✓ MLPClassifier trains iteratively since at each time step the partial derivatives of the loss function with respect to the model parameters are computed to update the parameters.

✓ It can also have a regularisation term added to the loss function that shrinks model parameters to prevent overfitting.

✓Main advantage - very fast and accurate.

✓My choice for such (fake) data

4. Results

- ✓ Implemented cross platform universal solution for cluster analysis, which allows end user see distribution histograms in raw time series, use different data preprocessing approaches and different clustering algorithms (5 and can be easily added), visualises results and saves resulted label data.
- ✓ Implemented cross platform solution for classification with 4 classification algorithms (one is the Deep Learning near network) for set of data series, which allow make a REST (Post) requests and receive the class of the sample.
- ✓ Checked my Data Science skills and received a lot of positive emotion from such interesting challenge. REALLY AMAZING!
- ✓ Repository link: https://github.com/StanislavKonchenko/Creos_Challenge/upload/main

5. Future work

✓ Obtain a position in Creos!

✓Allow user to add model options for clustering algorithms

✓ Add hyper-parameter turning options to classification trainer

✓ Install DevOps pipeline with automated CI/CD

6. Questions

✓ Go to the code!


THE END