# Introduction

The weather prediction is one of the oldest and most used predictions. When hundreds of years ago, the weather was predicted with the use of the gods, today there are very complex algorithms which determine if an umbrella is needed today. The KMNI has a free API weather database. It provides many features concerning the weather.

Weather predictions are made every day, and hence it would be highly labour intensive to do this manually. The challenge of this report, therefore mainly lays in the atomization of machine learning predictions using KubeFlow. This atomization should be accomplished while preserving some of the essential arguments which the model needs.

In this assignment, this database is used to train the model, and the latest information is used to predict the weather of today. In a production phase models are trained, validated, and the best performing model is selected. In the serving phase, a model can be chosen to use, and the weather of today will be predicted based on the data of yesterday’s weather.

The pipelines are uploaded to the GCP kubeflow pipeline UI and ran recurring with a periodic trigger. The deployment pipeline is running every week to retrain the model with the latest data. The serving pipeline is running every day to provide a prediction of today’s weather.

# Architecture

The general architecture consists of two pipelines, the deployment pipeline, which creates the models, trains and evaluates them. The models which are stored can be used to predict the weather using the serving pipeline. The results are saved in google cloud storage and visualized.

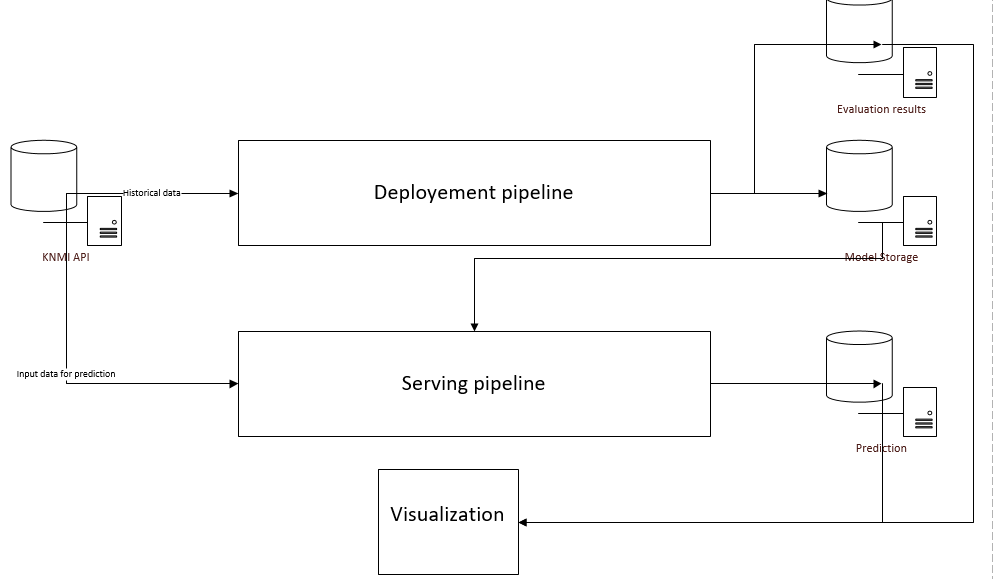
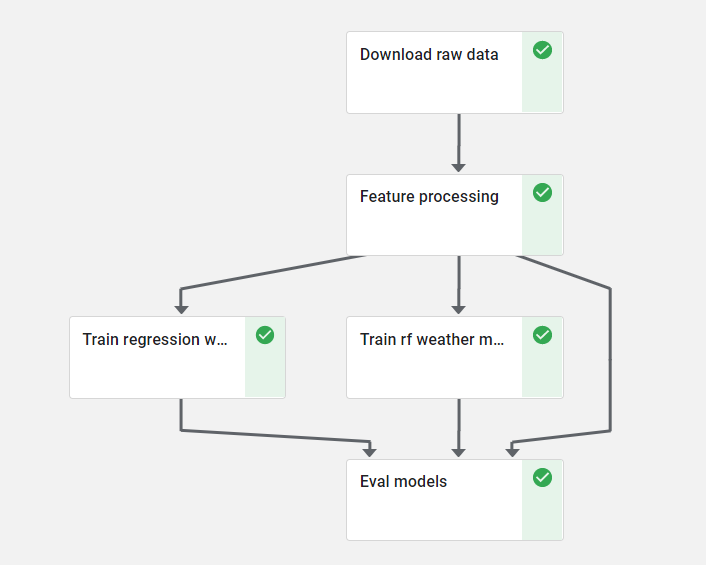


Figure 1: General architechture

## Deployment pipeline

This pipeline is created using the notebook “pipeline\_deploy.ipynb”. This notebook is converted into a python file which is then compiled into a tar.gz file. This file is uploaded in the GCP kubeflow pipeline UI and ran every week.

*Input arguments:*

*Validation days: number of rows used for validation of the model*

Download raw data:

This function downloads historical data from the KNMI API. This data is stored in a google cloud storage bucket.

Feature processing

The data is processed, and features with no or minimal variance are removed. There is also a new feature created. The “future\_TG”, this is the temperature shifted one day. Hence this is for that row the temperature of tomorrow. “future\_TG” will be our y value. Again the data is stored in a google cloud storage.

Train regression weather model & train rf weather model

In this container, models are trained with the resulting data from the previous container. This results in two trained models, linear regression and a random forest repressor. These are again stored in a google cloud storage.

Eval models

The models are validated and compared using MAE and MSE, advice for the best model is returned.

## Serving pipeline

The serving pipeline is created using the notebook “predict\_cont\_pipeline.ipynb”. This notebook is converted to a .py file and then to compiled to a tar.gz file which runs in kubeflow.

Download raw data

The pipeline downloads the weather data of yesterday. This data is then stored in a google cloud storage bucket.

Feature processing

The raw data which was downloaded in the first container is processed to make it fit our model.

Predict weather model

There is a prediction made based on the data from the feature processing container. This prediction is stored in the google cloud storage bucket. The pipeline returns the predicted weather for today.

# Deployment

# Challenges