MODEL PRESENTATION

TEAM 4:
SPITKOVSKA VLADYSLAVA
TYSCHENKO IVAN
ZASIADKO MATVIY
NYCH KATERYNA
HONCHARENCKO VLADYSLAV

Teacher:

Andrew Kurochkin

06.05.2024

Plan of presentation

- Introdusion, why our project is important
- How the project work:
    - Information gathering
    - Data collection and composition
    - NLP
    - PCA
- Model selection and tunning hyperparameters
- Problem-solution
- Deployment:
    - Backend
    - Frontend
- Demo
- Github repo

RUSSIAN-UKRAINIAN WAR

The Russian-Ukrainian War has persisted for over a decade. On February 24, 2022, the situation escalated from partial occupation to full-scale invasion. Millions of Ukrainians have joined the army or volunteered in support organizations within Ukraine and beyond its borders. Millions have lost their homes and families. Russia, as the aggressor state, has violated the rights of every one of us as individuals.

* Mariupol under the occupation

*Ruined by rocker civil house in Dnipro

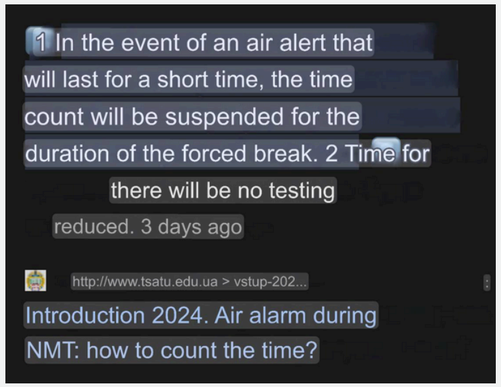
Ukrainian soilders near the flag of Ukraine

*First photos after Bucha’s occupation

LIFE IN UKRAINE IN THE WAR

Continuing civilian life in Ukrainian cities, the availability of air raid alerts is crucial. The presence of air raid alerts can disrupt daily routines not only for individuals but also negatively impact Ukrainian businesses, the organization of public events, the conduct of nationwide examinations, and more.

*Air Alarms map (almost all regions are under the attack)



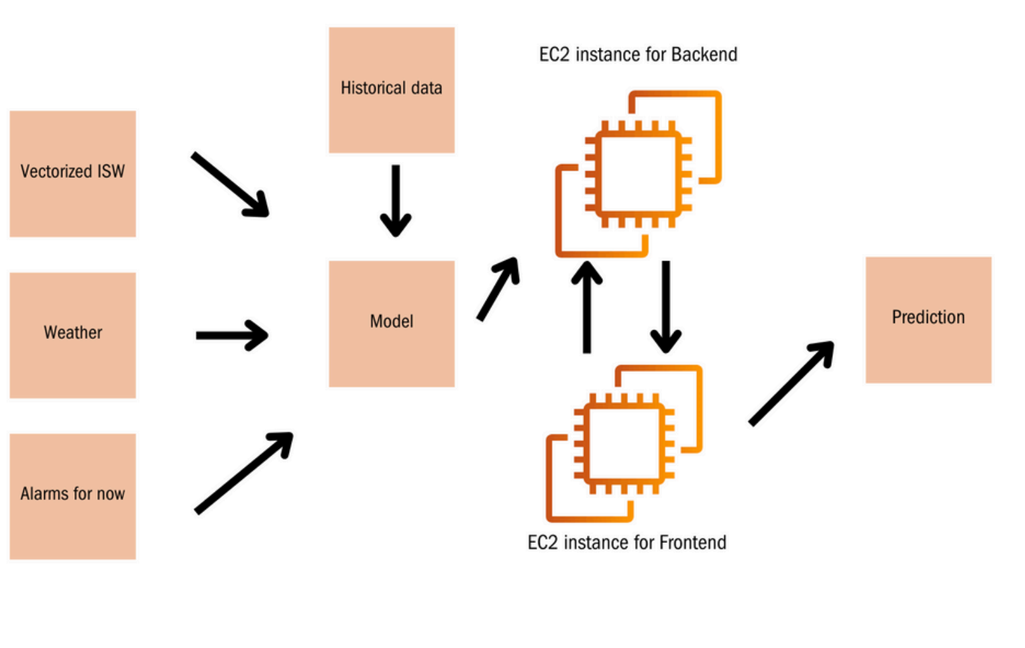
*Ukrainian school-leavers are not able to continue their exam passing if there's an air alarm d

*Children in school having a lessonin the shelter during air alarm

*People waiting for air alarm ending in metro (metro doesn't work during it)

HOW CAN WE BE USEFUL?

This led to the idea of creating a web service that would predict such information regionally for the following (day/12 hours). Modern technologies and available historical data allow the creation of a model that analyzes past events and, following a specific algorithm, forecasts likely future events.



*Simple schema of our system work

Whats **Secure Sky**?
An API for getting information about air alarms state for any region of Ukraine.

SECURE SKY

API - tands for application programming interface, which is a set of definitions and protocols for building and integrating application software.

Model - a program that can find patterns or make decisions from a previously unseen dataset

How does it work?

INFORMATION GATHERING (1/3)

The first step involves determining the content of our future dataset - fields of events favorable or unfavorable for alerts. The following questions make sense:

- **What data could be informative?**
- **What sources can be used to obtain this information?**
- **What types of data might appear in the datasets?**

dataset - is a collection of data. To visualise the structure we use tables

	A	B	C	D	E	F	G
1	Date	Average price in USD	Total Sold	Smal Avocados Sold	Large Avocados Sold	Extra Large Avocados Sold	City
2	12/27/2022	1.29	319746	292097	27351	298	Atlanta
3	12/20/2022	1.38	272580	251774	20702	103	Atlanta
4	12/13/2022	1.26	356227	324932	31019	276	Atlanta
5	12/6/2022	1.37	306947	283024	23741	182	Atlanta
6	11/29/2022	1.29	279486	250289	28890	308	Atlanta
7	11/22/2022	1.3	300032	269799	29732	501	Atlanta
8	11/15/2022	1.43	292814	263916	28442	456	Atlanta
9	11/8/2022	1.42	285413	250441	34483	488	Atlanta
10	11/1/2022	1.29	353690	290458	62980	253	Atlanta
11	10/25/2022	1.39	301727	236814	64608	304	Atlanta
12	10/18/2022	1.39	288733	228710	59732	291	Atlanta
13	10/11/2022	1.25	347580	255933	91047	600	Atlanta
14	10/4/2022	1.26	359222	265797	92780	644	Atlanta
15	9/27/2022	1.07	324659	262107	61871	681	Atlanta

link

data type - an attribute associated with a piece of data that tells a computer system how to interpret its value.

Common data types

Data type:	Example value:
Integer	35462216
Floating-point	0.002756
Char	H
String	Hello, World!
Boolean	true

link

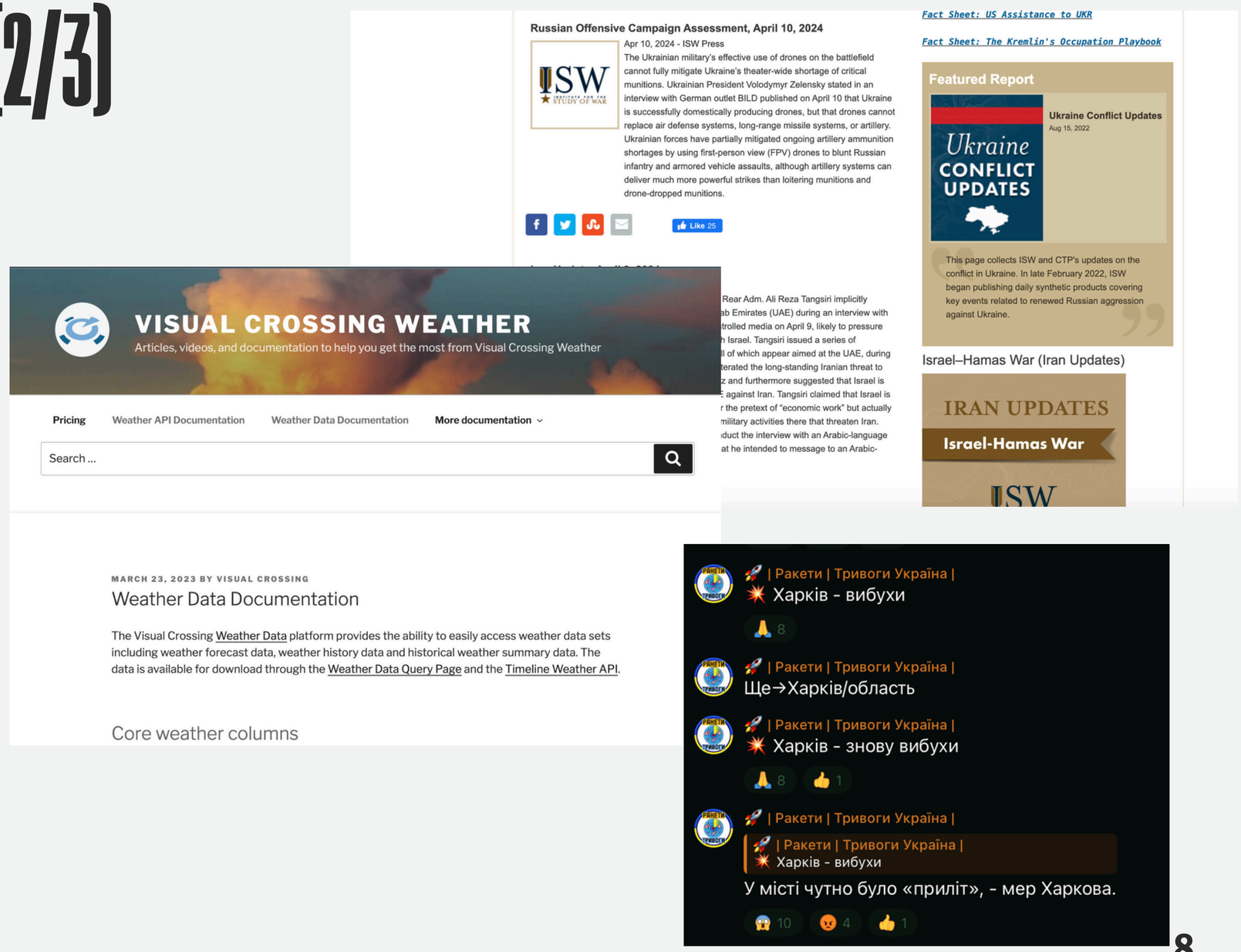
INFORMATION GATHERING (2/3)

Three sources of information were selected:

ISW(Institute for the Study of War), which daily records historical information about the war in Ukraine from 02/25/2022 to the present. For training our future model, we will take only a portion of the records for 1.5 years.

Visual Crossing - API with weather information. We will obtain it regionally for the same period.

Telegram channels and APIs with information about the presence of air raid alerts.



INFORMATION GATHERING (3/3)

Step 1: **Write code to retrieve information from each source separately.**

Step 2: **Obtain and analyze information from each dataset. Conduct EDA for each dataset. Identify correlations regarding parameters.**

Step 3: **Normalize the information in all datasets.**

Step 4: **Merge/concat the datasets into the final one. Conduct EDA of the final dataset.**

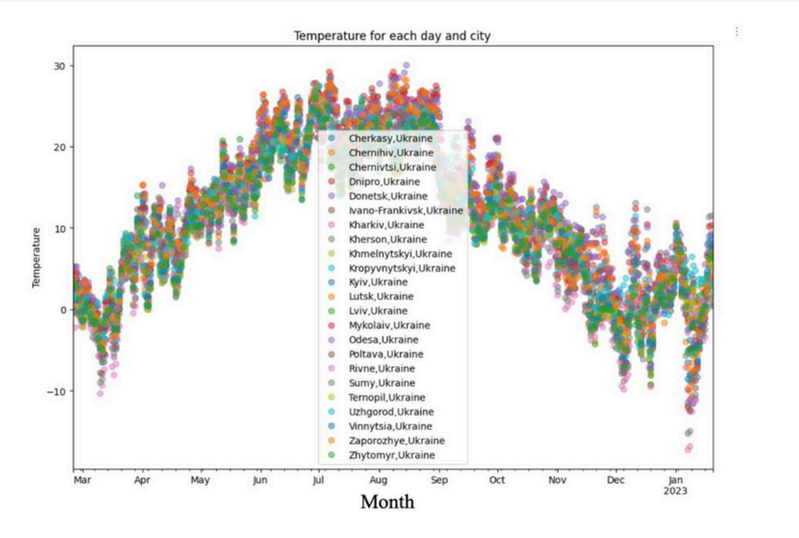
Normalization - reorganizing dataset to remove any unstructured or redundant data to enable a superior, more logical means of storing that data.
EDA - Exploratory data analysis is used to analyze data sets and summarize their main characteristics, employing data visualization methods.
Concatenating - is a combining of two or more data sets, one after the other, into a single data set.
Merging - a process of combining two or more similar records into a single one.
Correlation - a functional mathematical relationship among the parameters

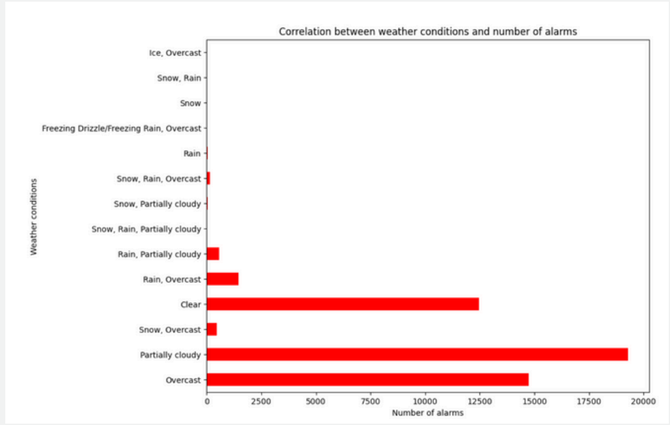
DATA ANALYSIS AND COMPOSITION (1/4)

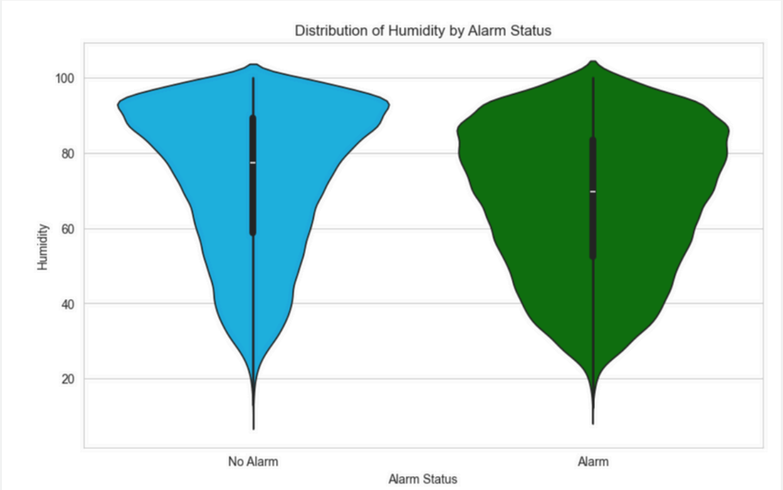
At this stage, we have created 3 datasets for 3 separate sources. It is worth noting separately that besides API, we also extracted information about alerts from Telegram.

DATA ANALYSIS AND COMPOSITION (2/4)

During the correlation analysis stage, we delved deeper into the specifics of our data individually. Based on this, we added a series of features.







DATA ANALYSIS AND COMPOSITION (3/4)

During data normalization, even before the merging stage, we separately conducted NLP and PCA for the ISW dataset.

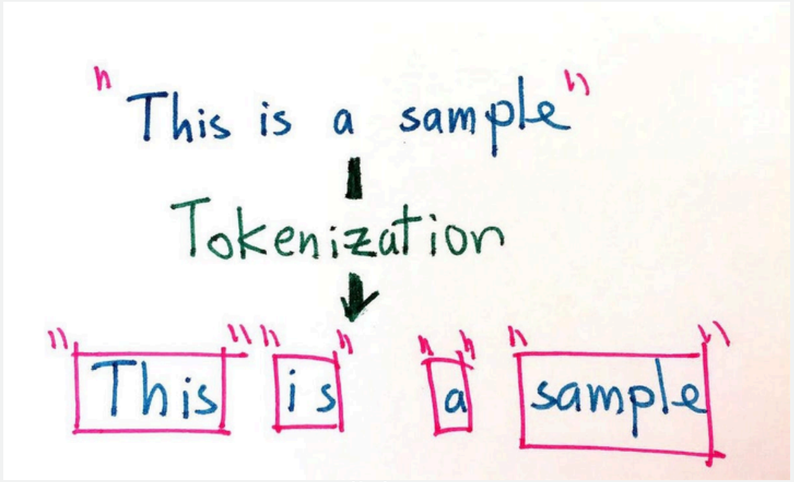
NLP

Natural language processing (NLP) is the application of computational methods to not only extract information from text but also model different applications on top of it. All language-based text has systematic structure or rules, which are often referred to as morphology, for example, the past tense of “jump” is always “jumped”.
We can divide this into several stages:

*tokenization – Splitting text into smaller units such as words or phrases.

STAGE 1 - TOKENIZATION

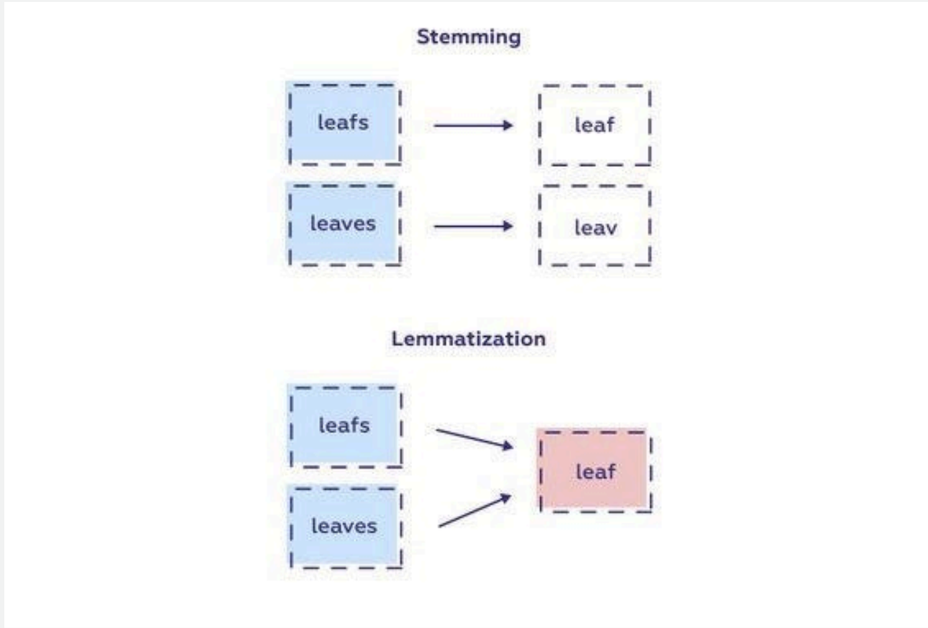
The task of segmenting text into relevant words is called tokenization.
In its simplest form, tokenization can be achieved by splitting text using whitespace.



link

STAGE 2 - STEMMING OR LEMMATIZATION:

The task of reducing each word to its root. For example, “Walk” is the root for words like “Walks”, “Walking”, “Walked”. Usually, the root may hold significantly more meaning than the tense itself. So in NLP tasks, it’s very important to extract the root for the words in the text.



link

NLP

Text vectorization is the process of converting text into numerical representations (or “vectors”) that can be understood by ML models. It transforms unstructured text into structured numeric data with the goal to represent the semantic meaning of text in a mathematical format.

STAGE 3: VECTORIZATION.

Here's an example of vectorized text:

	about	all	cent	cents	money	new	old	one	two
doc	1	1	3	1	1	1	1	1	1

In theory (a)

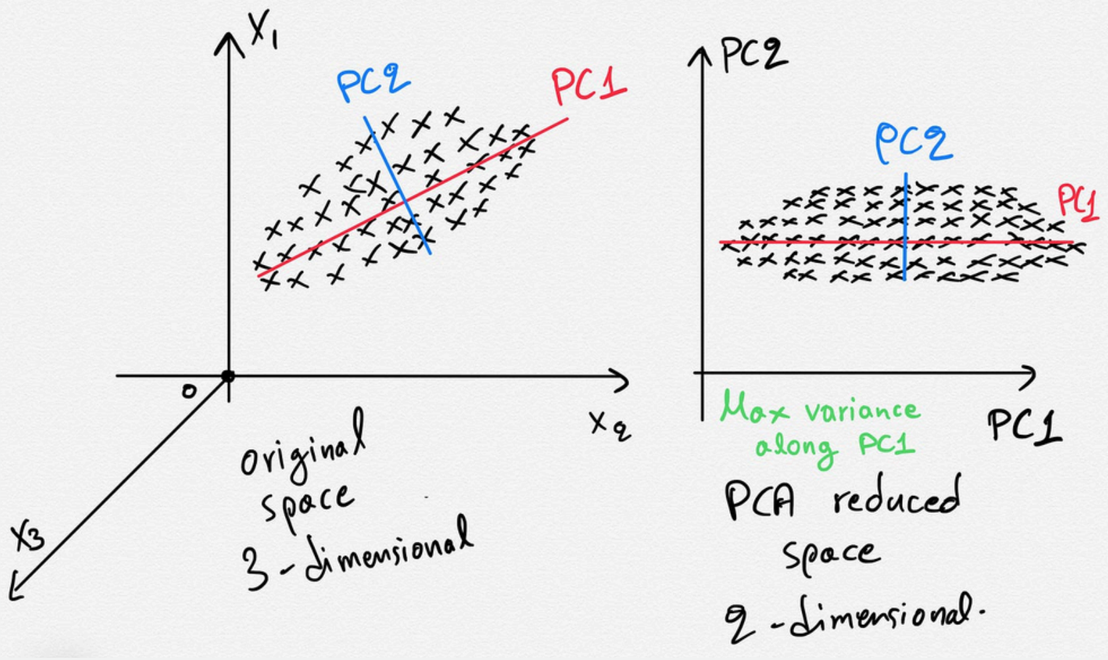
Index	0	1	2	3	4	5	6	7	8
doc	1	1	3	1	1	1	1	1	1

In practice (b)

link

PCA

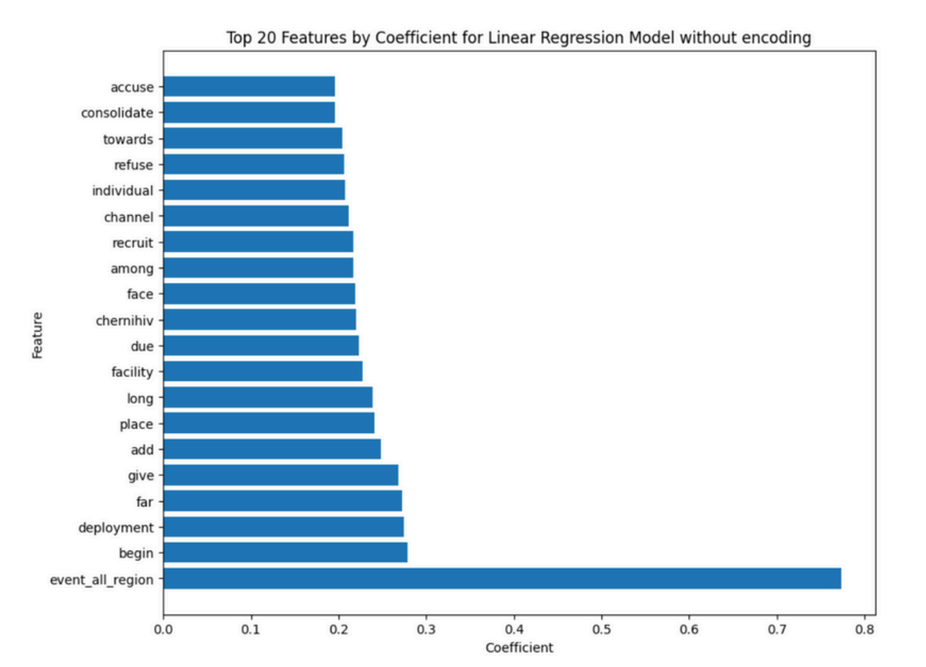
Principal Component Analysis is a statistical method that transforms high-dimensional data into a lower-dimensional form while preserving the most important information. It accomplishes this by identifying new axes, called principal components, along which the data varies the most.
The main concept, as I seem to have already mentioned, is the creation of new indices and merging certain parameters into one. (Example: height column + weight column = person parameters column)

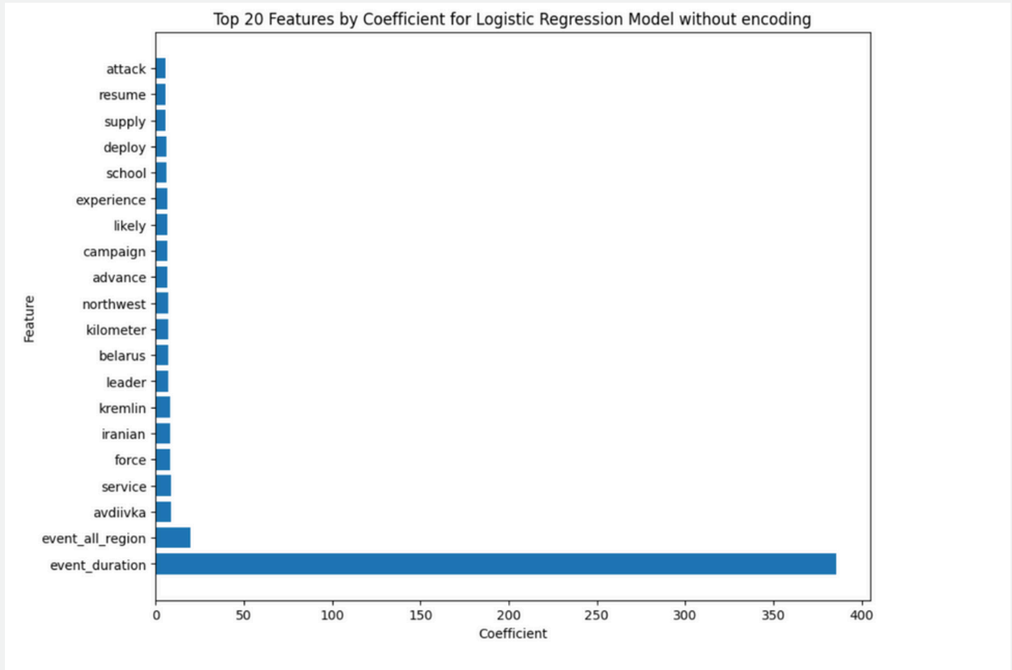


*simple PCA visualization

DATA ANALYSIS AND COMPOSITION (4/4)

We then merged the datasets into one large dataset and removed columns (attributes) that had high correlation coefficients with the is_alarm column.
After all our dataset has 754 columns and weights 1512 MB.





*EDA parts before dropping the several odd columns

MODEL SELECTION ☆

THE THEORETICAL PROCESS OF MODEL SELECTION INVOLVES THE FOLLOWING STEPS:

DEFINE THE PROBLEM

Clearly define the problem and determine whether it is a classification or regression problem, and identify the performance metrics to evaluate the models.

SELECT MODEL CANDIDATES

Identify a set of candidate models that are suitable for the problem.

PARTITION THE DATA

Split the available data into training, validation, and test sets.

TRAIN MODELS

Fit each candidate model to the training data using an appropriate algorithm.

EVALUATE MODELS

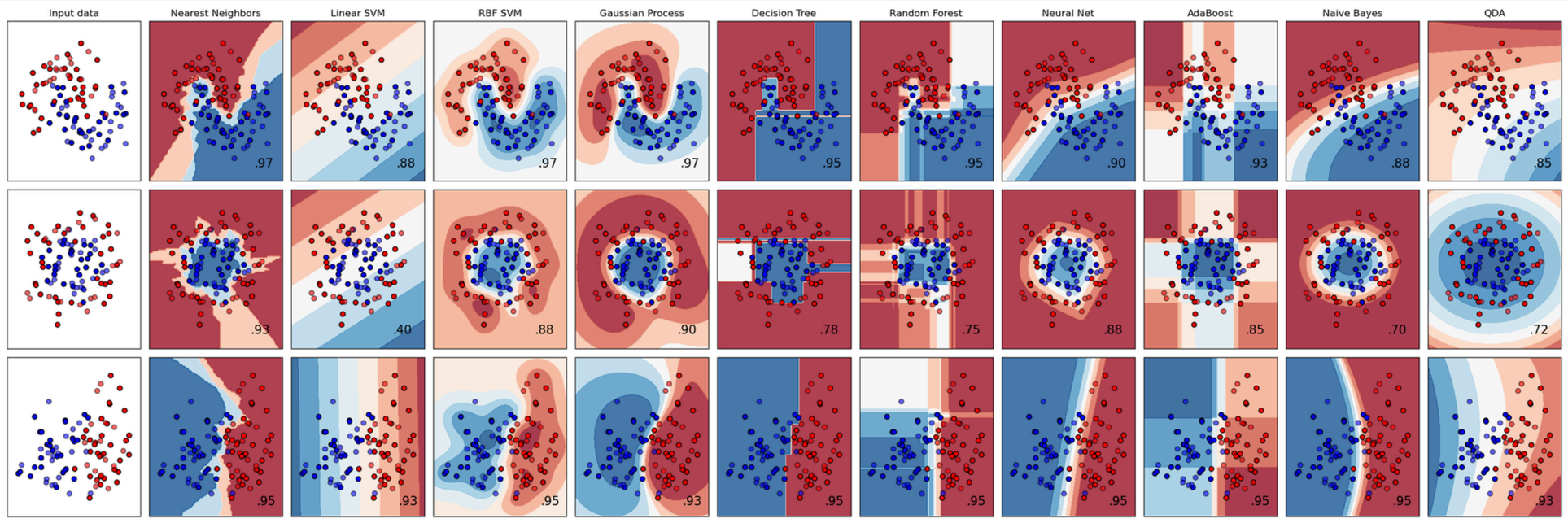
Assess the performance of each model on the validation set using predefined metrics such as accuracy, precision, recall, or mean squared error.

VALIDATE THE FINAL MODEL

Use the test set to validate the performance of the selected model.

Model selection (1/3)

When choosing a model for data analysis, it is crucial to consider the nature of the data and the expected outcome. For instance, data with numerous features and complex interactions between them may benefit from the use of deep learning methods, such as neural networks.

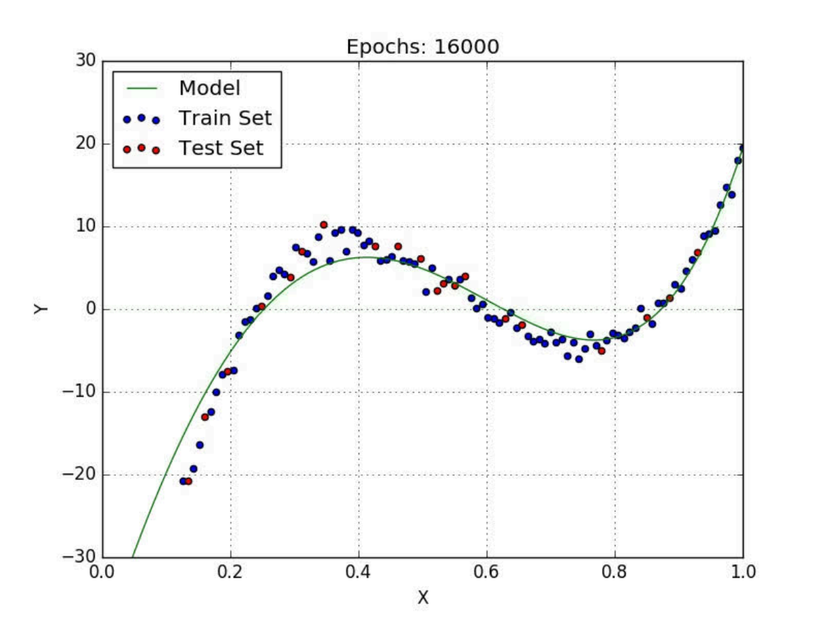


*how do models work with different data

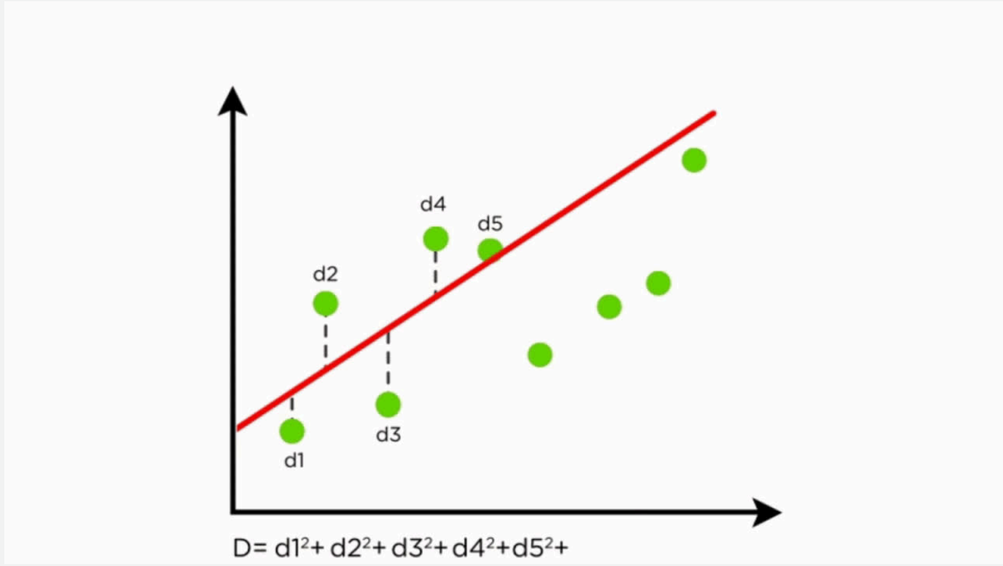
Model selection (2/3)

Some of the simplest and most interpretable models include logistic and linear regression. However, achieving the highest possible model accuracy for any given dataset using typical or even the most primitive models is not always feasible. Taking into account various aspects identified in earlier stages of analysis, we proceeded with selecting the best models.

*Logistic regression visualization

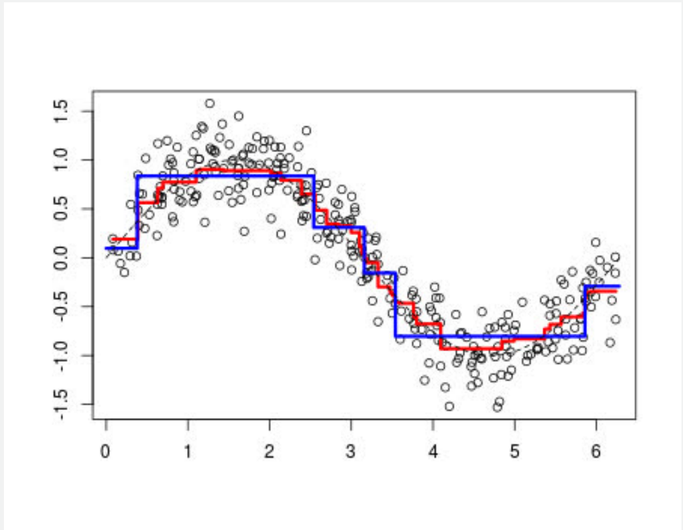


*Linear regression visualization



Model selection (3/3)

link



*XGBoost visualization

Based on the characteristics of our data and its structured nature, we chose the XGBoost model for analysis. XGBoost is an effective machine learning algorithm based on decision tree search and gradient boosting. This algorithm provides high prediction accuracy in a short time with minimal computational resources.

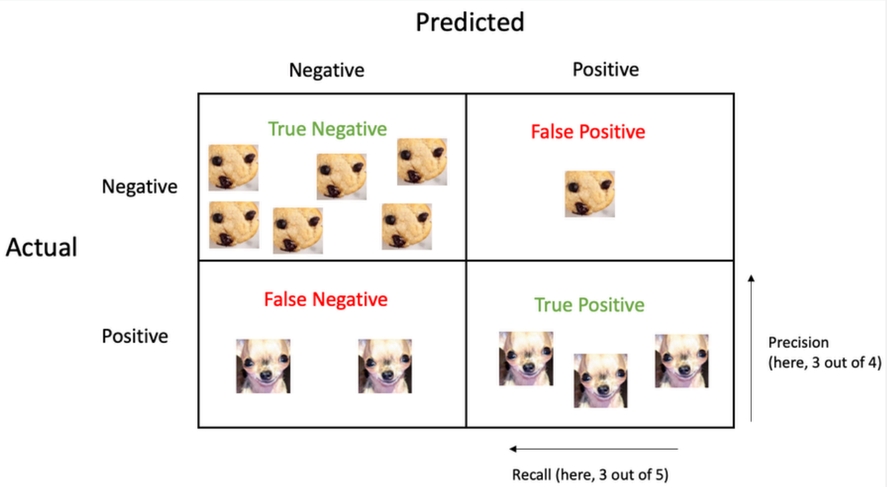
*gradient boosting - is a machine learning ensemble technique that combines the predictions of multiple weak learners, typically decision trees, sequentially.

EVALUATION METRICS

Among the popular metrics are: accuracy, precision, recall, MSE (Mean squared error), etc. They all have certain mathematical formulas under the hood. We checked the most widely used of them.

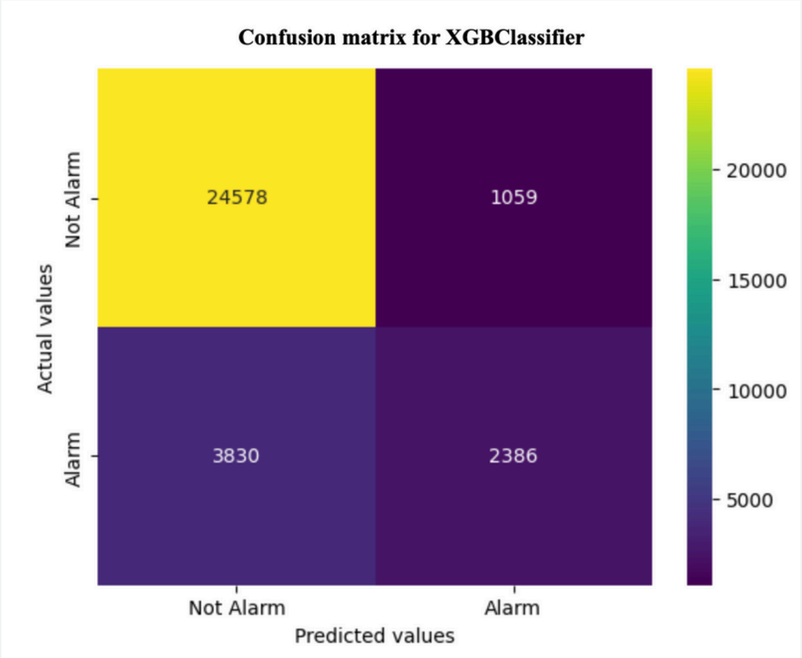
Metric	Formula
True positive rate, recall	$\frac{TP}{TP+FN}$
False positive rate	$\frac{FP}{FP+TN}$
Precision	$\frac{TP}{TP+FP}$
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
F-measure	$\frac{2 \cdot precision \cdot recall}{precision + recall}$

link



link

A visually informative metric is the confusion matrix, which shows how exactly the model "guessed" the target values.

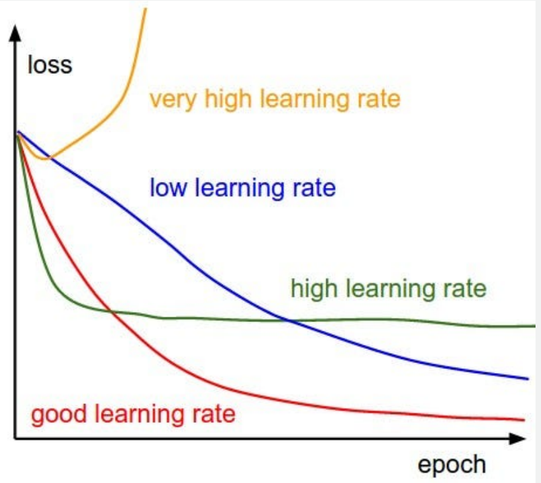


HYPERPARAMETER TUNING

XGBoost:

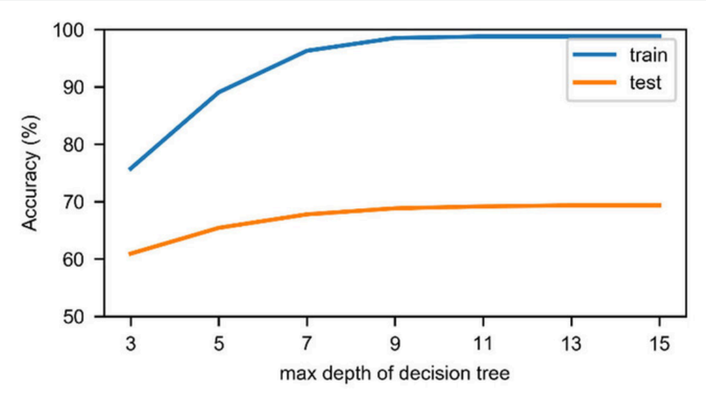
n_estimators=200, max_depth=10, learning_rate=0.4, max_delta_step=1, gamma=0, objective='binary:logitraw'

LEARNING RATE
It controls the step size during the training process



link

MAX DEPTH
In neural networks, the architecture is defined by the number of hidden layers and units in each layer.



link

N_ESTIMATORS
This parameter determines how deep the algorithm is allowed to build the tree. The deeper the tree gets the more resolution the tree will learn - as well as noise.

link

Problem-solution

Problems with ec2 instance

We had ptoblems with ec2 instance capacity, before it was t2.micro, than we decided to enlarge it to t2.small and it works well.

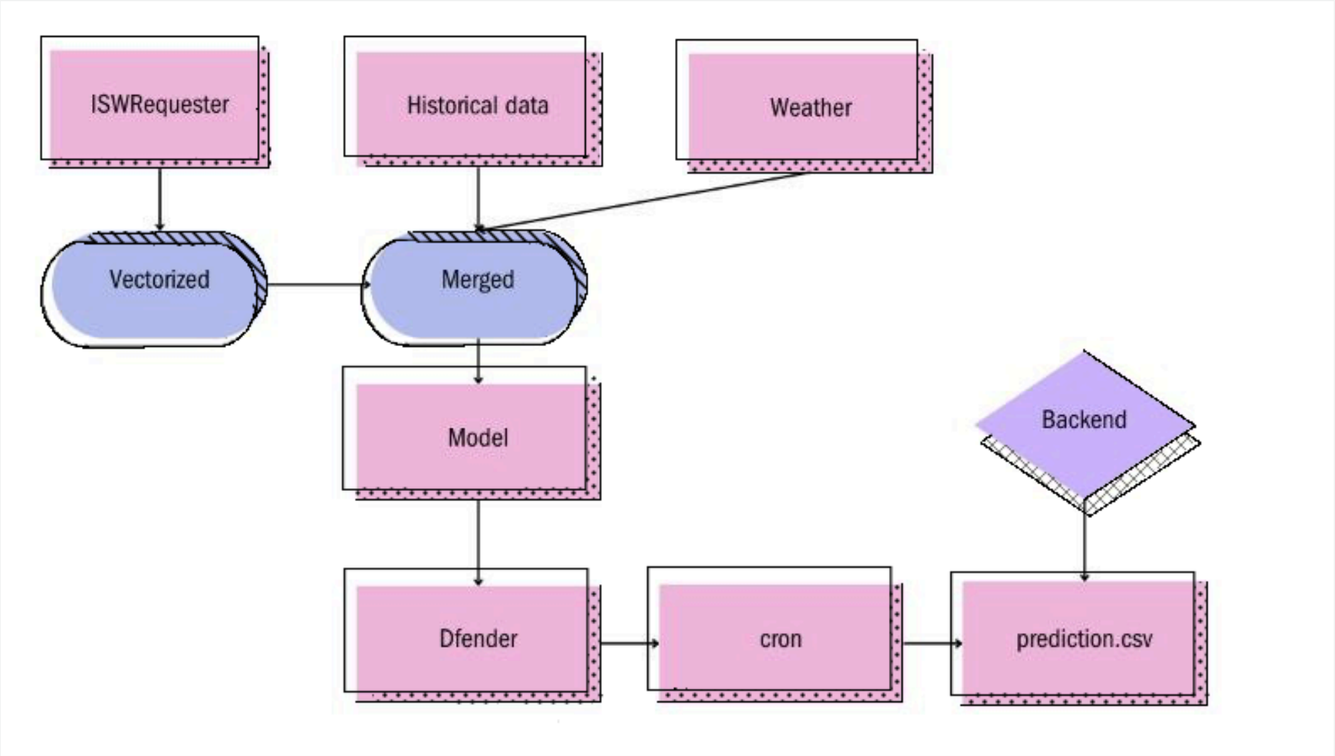
link

DEPLOYMENT

BACKEND (1/3)

API

First of all, we collect data from ISWRequester vectorized it, collect weather data, merge datasets with historical data that was provided. Second step was to train models on collected data and than our team use class Dfender to do prediction. We used cron to update data.

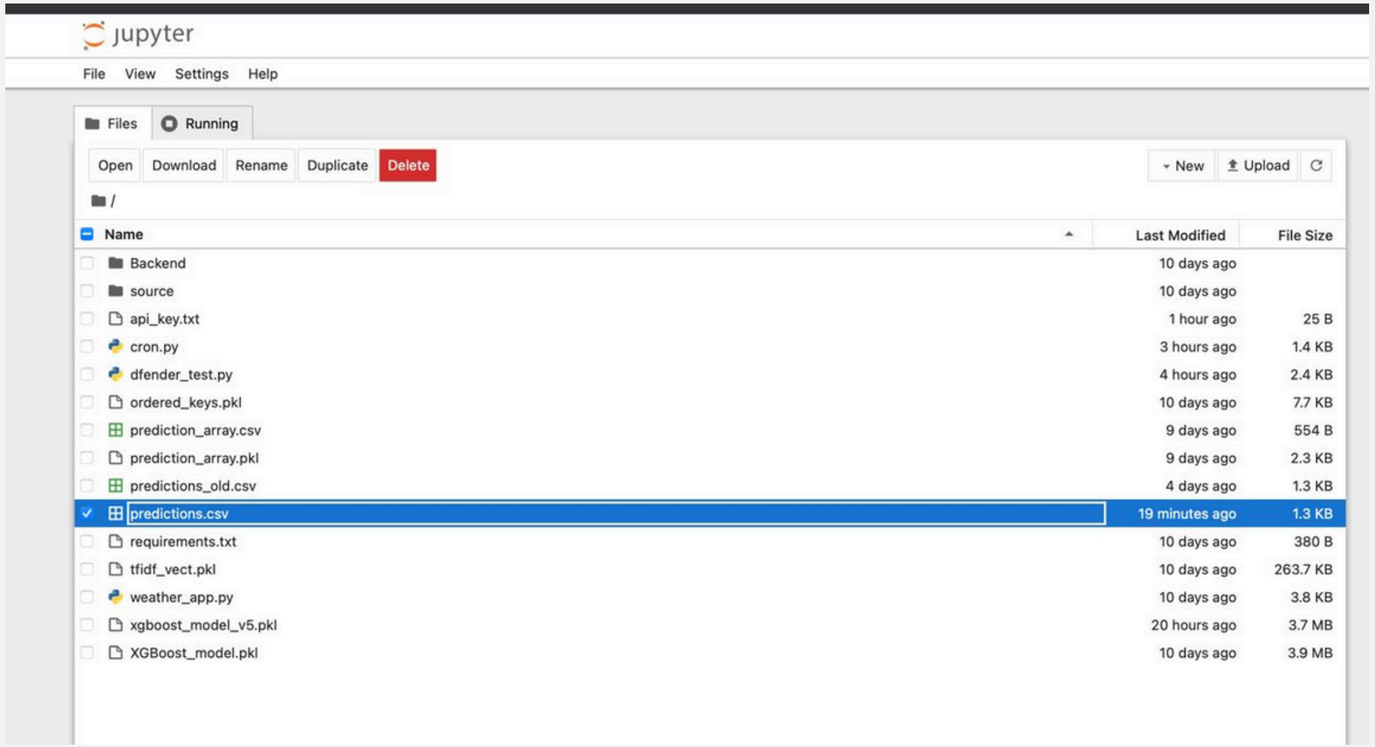


*System design

Backend (2/3)

Endpoint (backend.py)

When user do request prediction system, code endpoint is running on ec2 instance in jupyter notebook. Request goes to server, where turns into prediction.

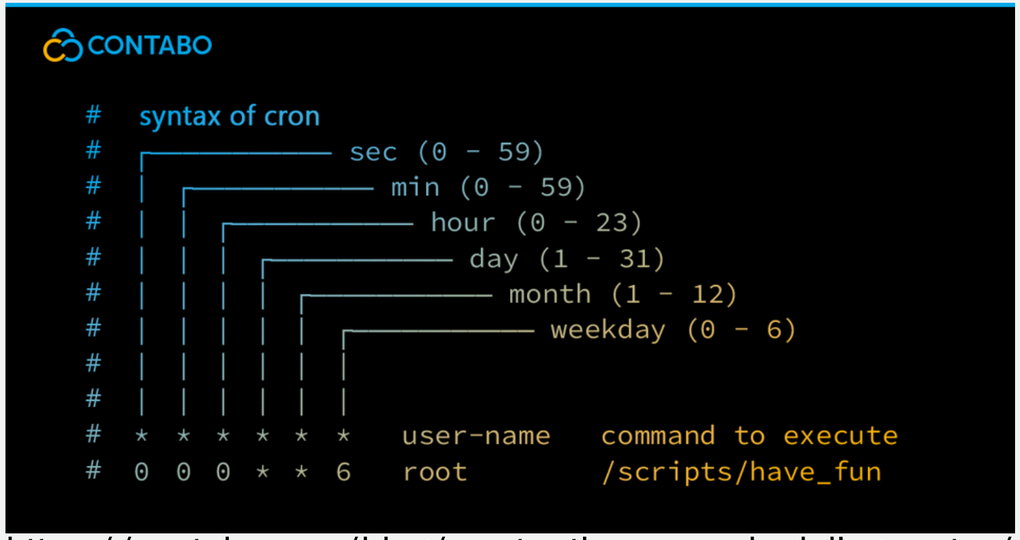


Backend (3/3)

cron.py

alarms.py

defender_test.py



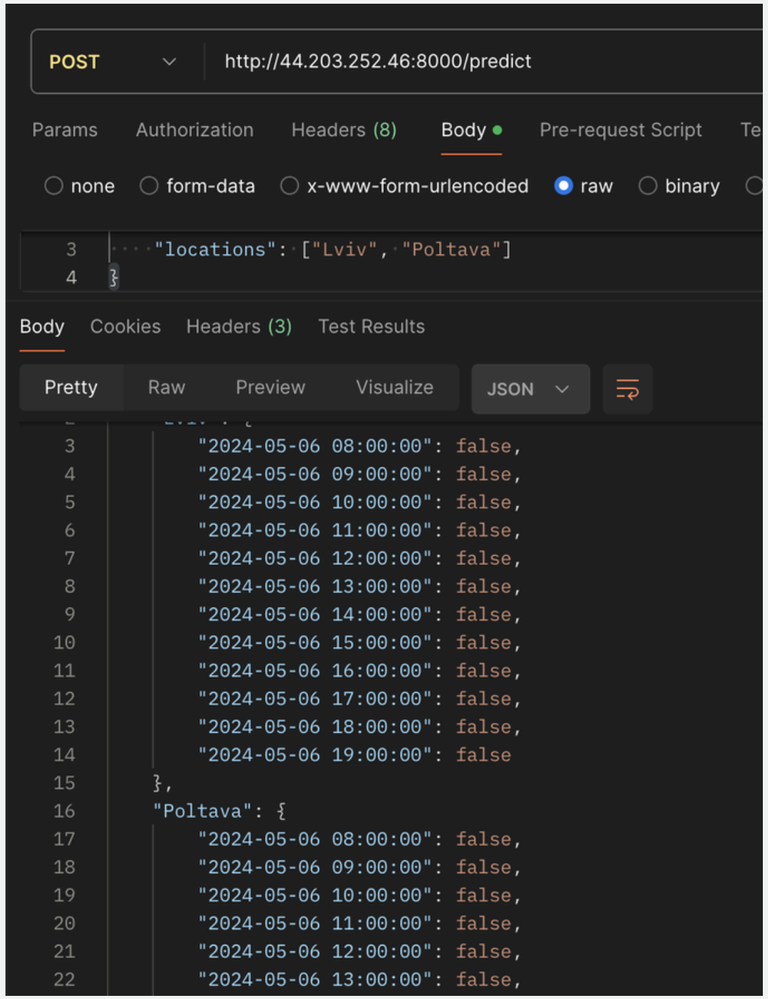
https://contabo.com/blog/master-the-cron-scheduling-syntax/

Results before frontend

The example of output that user receive from postman.

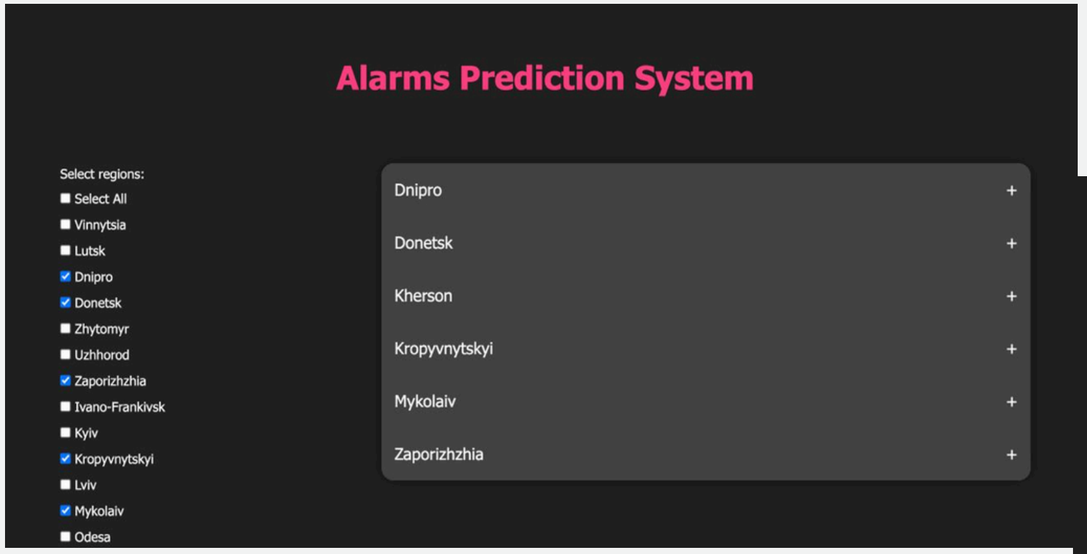
We get prediction for the next 12 hours for the 2 selected regions.

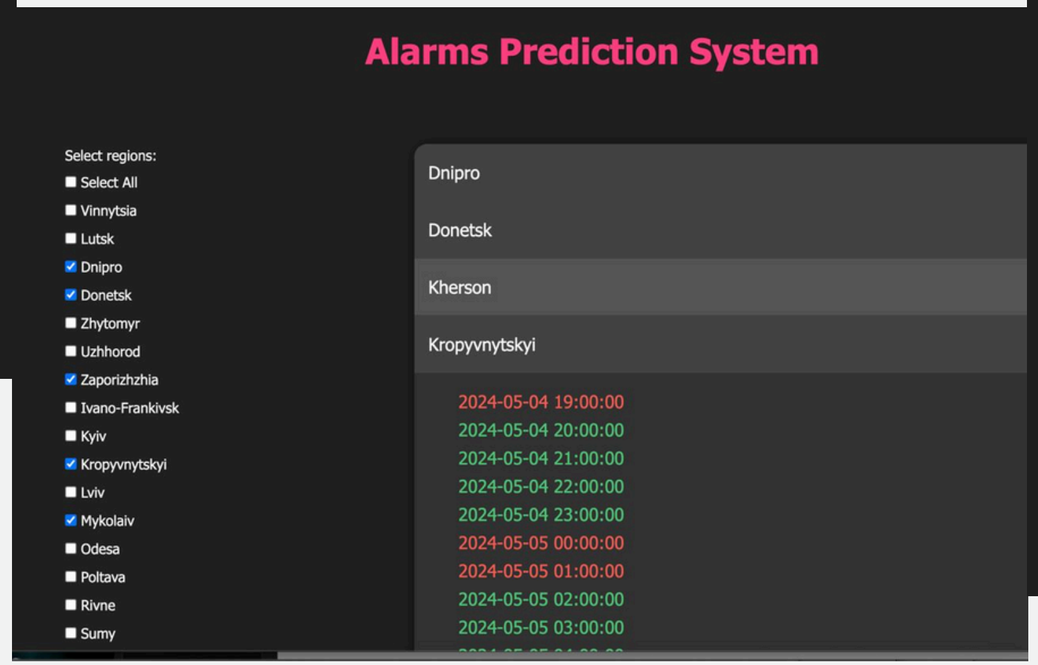
In this example we see predictions for Kyiv and Lviv

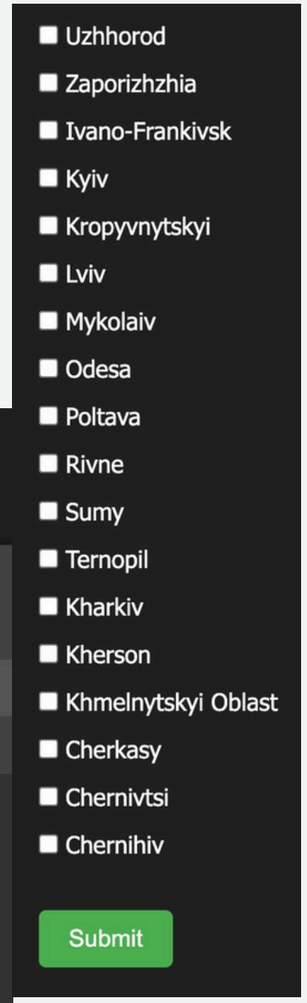


Frontend

Finaly we desided to do frontend. Here some exaples how it looks like:



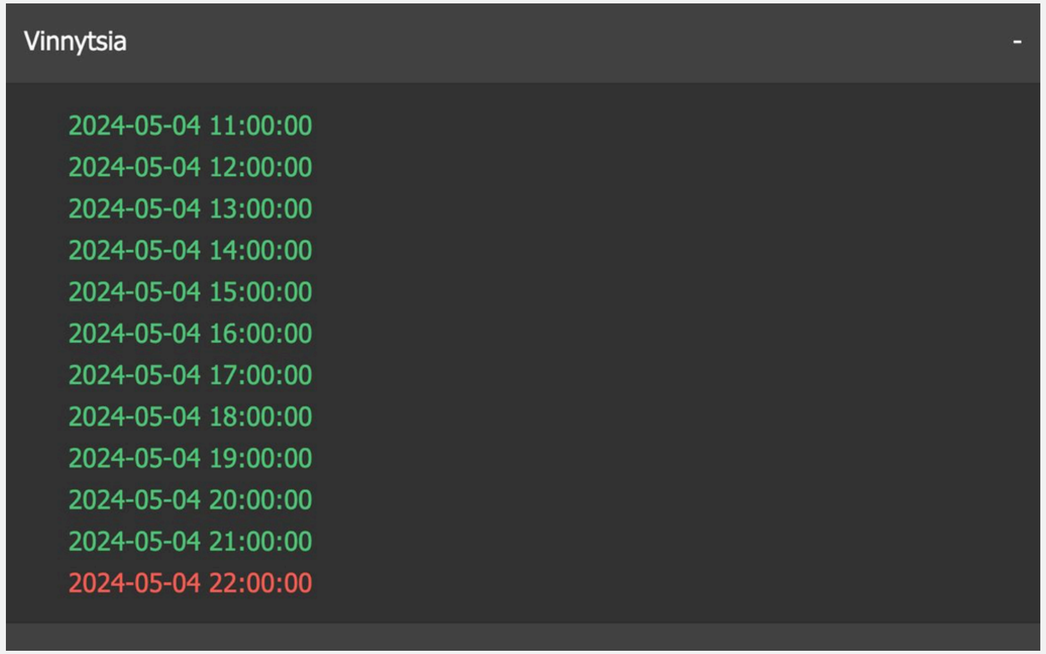




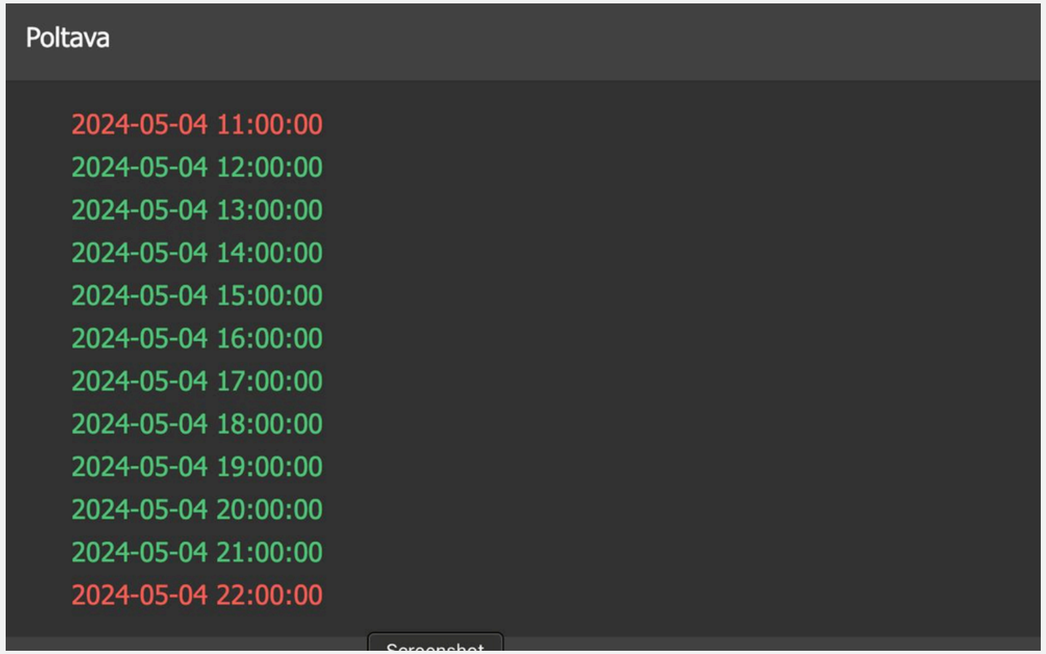
Demo/results of our work(1/2)

Some examples of system work

Example of prediction for Vinnytsia



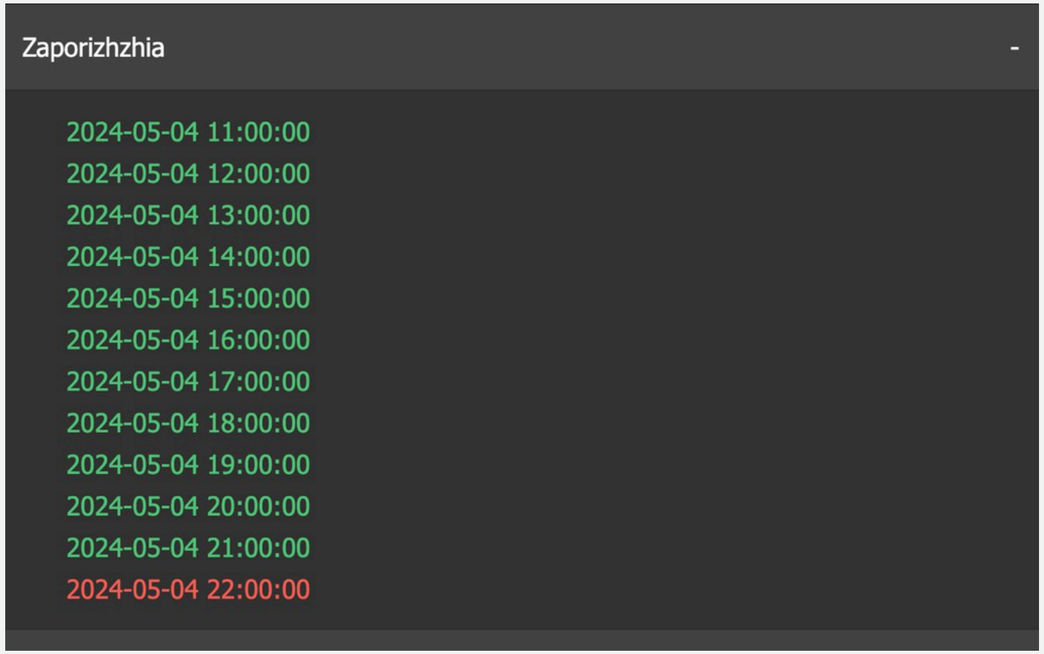
Example of prediction for Poltava



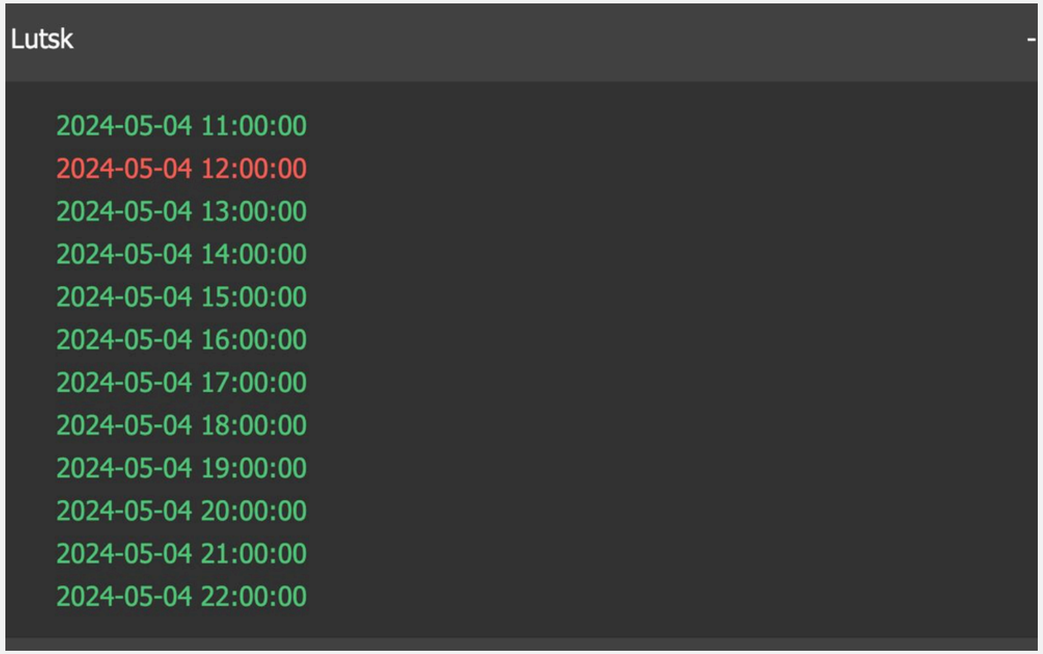
Demo/results of our work(2/2)

Other examples of system work

Example of prediction for Zaporizhia



Example of prediction for Lutsk



Git-repo:

https://github.com/synthco/4-pythonic-squad

Thank you!