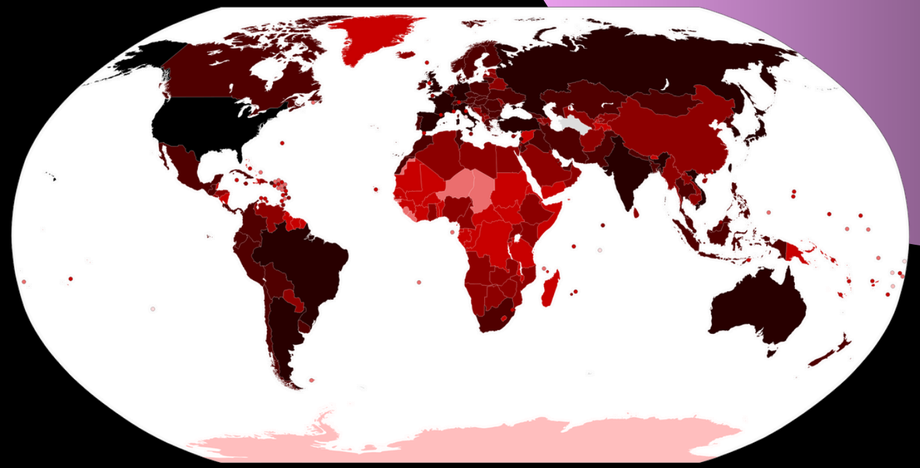
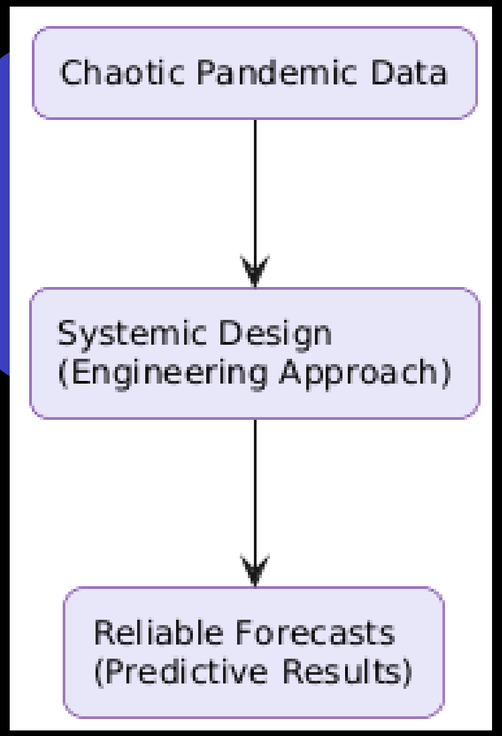
Covid 19 Forecasting Model design

Introduction



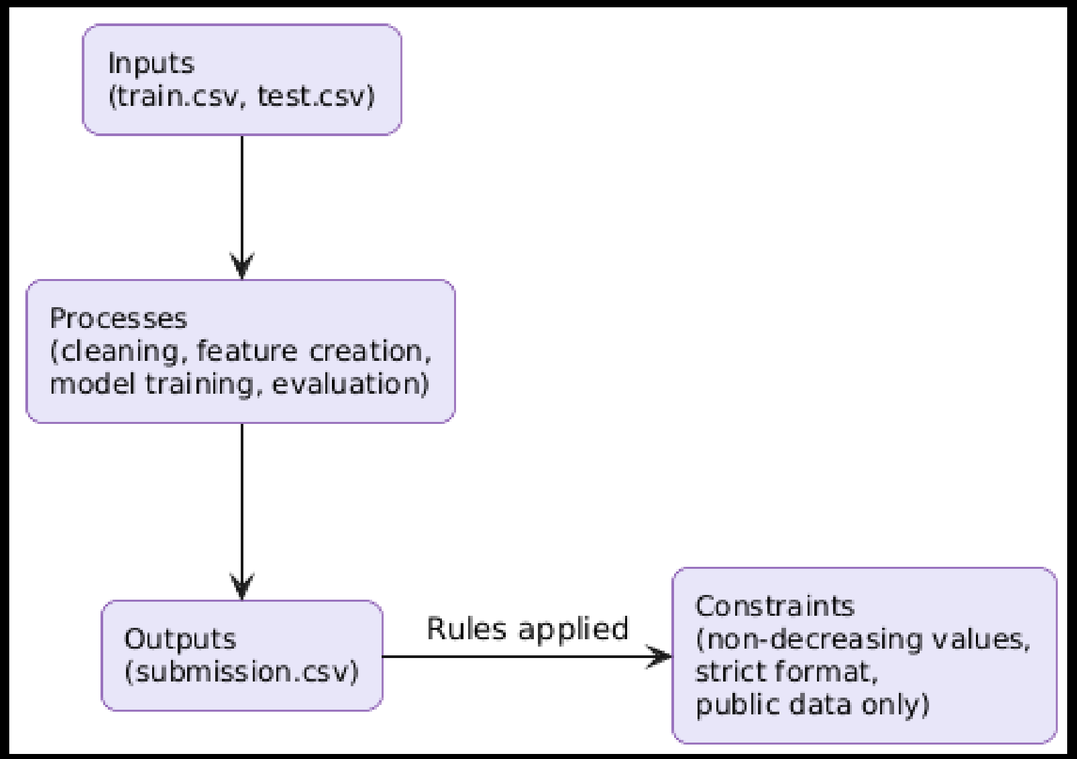
- The COVID-19 pandemic created a chaotic and unpredictable environment, where daily data changed rapidly and inconsistently.
- To address this, the Kaggle “COVID-19 Global Forecasting” competition challenged participants to design reliable forecasting systems.
- The main goal: predict confirmed cases and fatalities using global data under uncertainty.
- The system we developed applies systems engineering principles to handle chaos, randomness, and data instability.

PURPOSE AND MOTIVATION



- Many COVID-19 models focused only on accuracy but ignored reproducibility and adaptability.
- Our project aimed to build a modular and adaptive pipeline that could:
  - Process inconsistent data,
  - Adjust to sudden changes, and
  - Maintain stable predictions despite chaos.
- The system was designed not to avoid uncertainty, but to operate effectively within it.

SYSTEM COMPONENTS AND CONSTRAINTS



Main Components:

- Inputs: train.csv, test.csv → historical and future data.
- Processes: preprocessing, feature creation, model training, evaluation.
- Outputs: submission.csv → cumulative confirmed cases and deaths.

Main Constraints:

- Predictions must be non-decreasing and cumulative.
- Strict submission format required by Kaggle.
- Only publicly available data allowed.
- These constraints ensured fairness but increased system sensitivity to data noise.

SYSTEM CONTEXT: CHAOS AND UNCERTAINTY

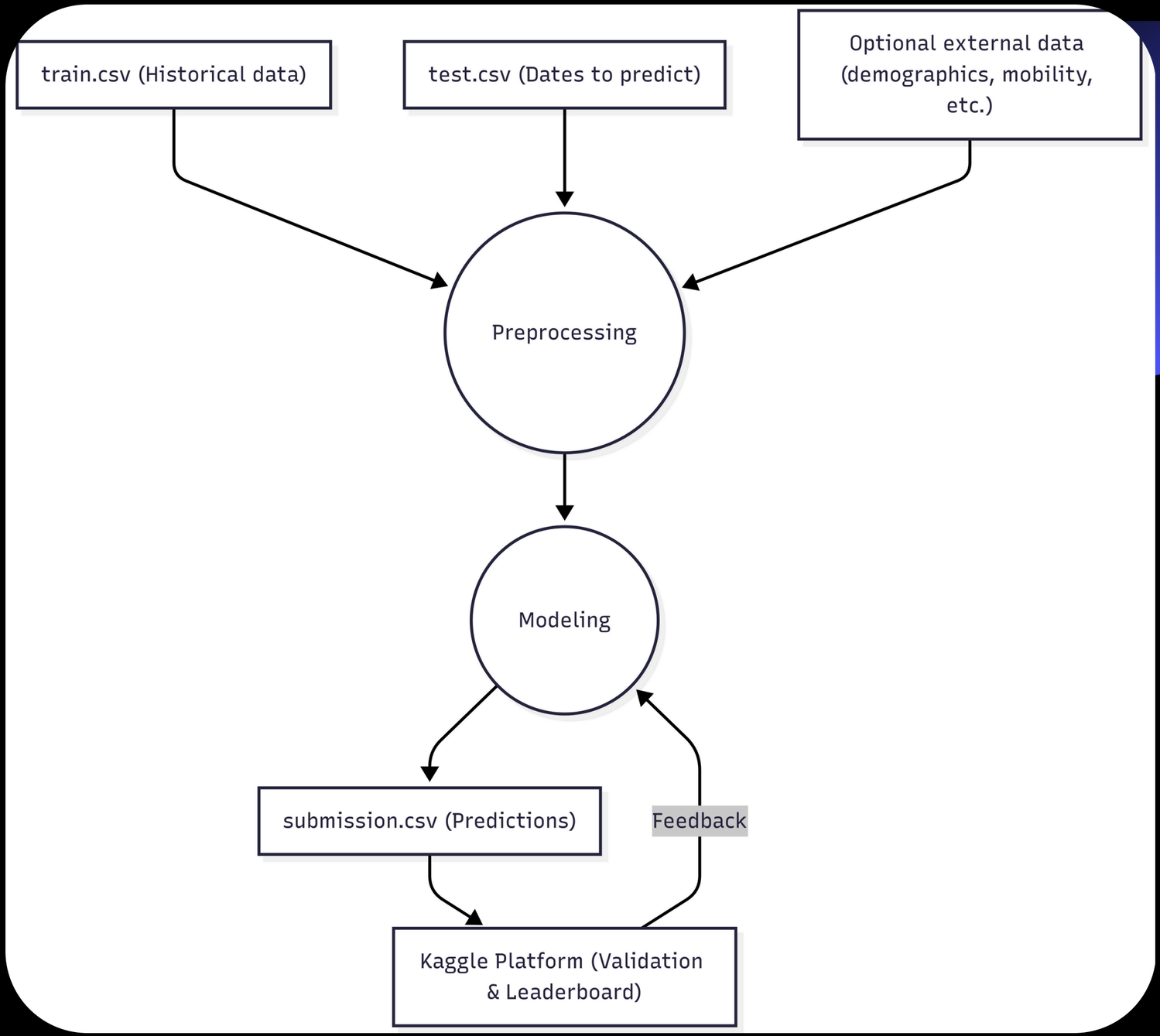
- Real-world data was irregular, delayed, or incomplete.
- Government actions (lockdowns, reopenings) and social behavior constantly changed the trend.
- Random reporting errors produced large shifts in the curves.
- The system had to stay stable and reproducible under these unpredictable dynamics.

Thynk
Unlimited.

Home

SYSTEM ANALYSIS

- Inputs
- Processes
- Outputs
- Actors



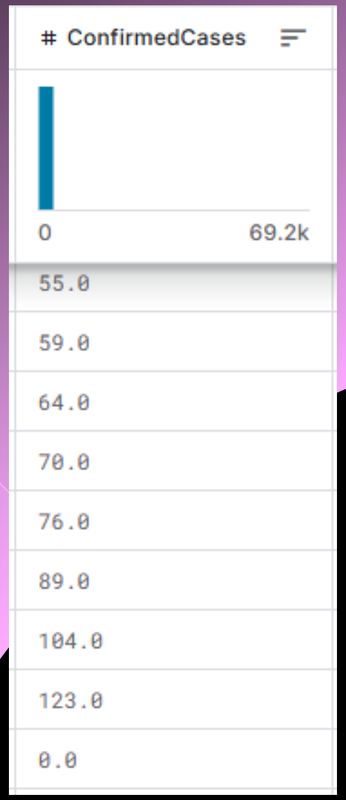
COMPLEXITY, SENSITIVITY AND CONTROL

Complexity

- Incomplete or noisy data
- Overfitting risk
- Hyperparameter sensitivity

Sensitivity

Few inconsistent data points or outliers can lead to large differences in predicted confirmed cases or fatalities

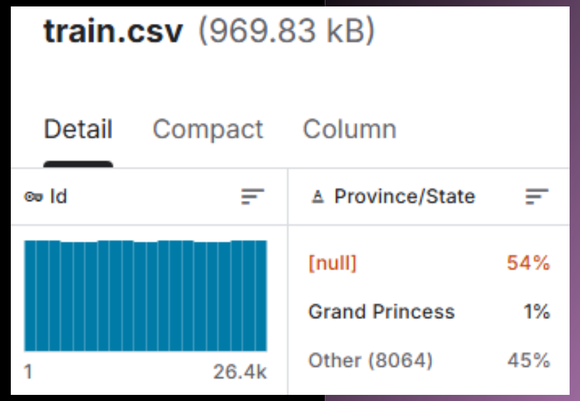


Control Mechanism

Algorithm 1 COVID-19 Forecasting and Feedback Control Loop

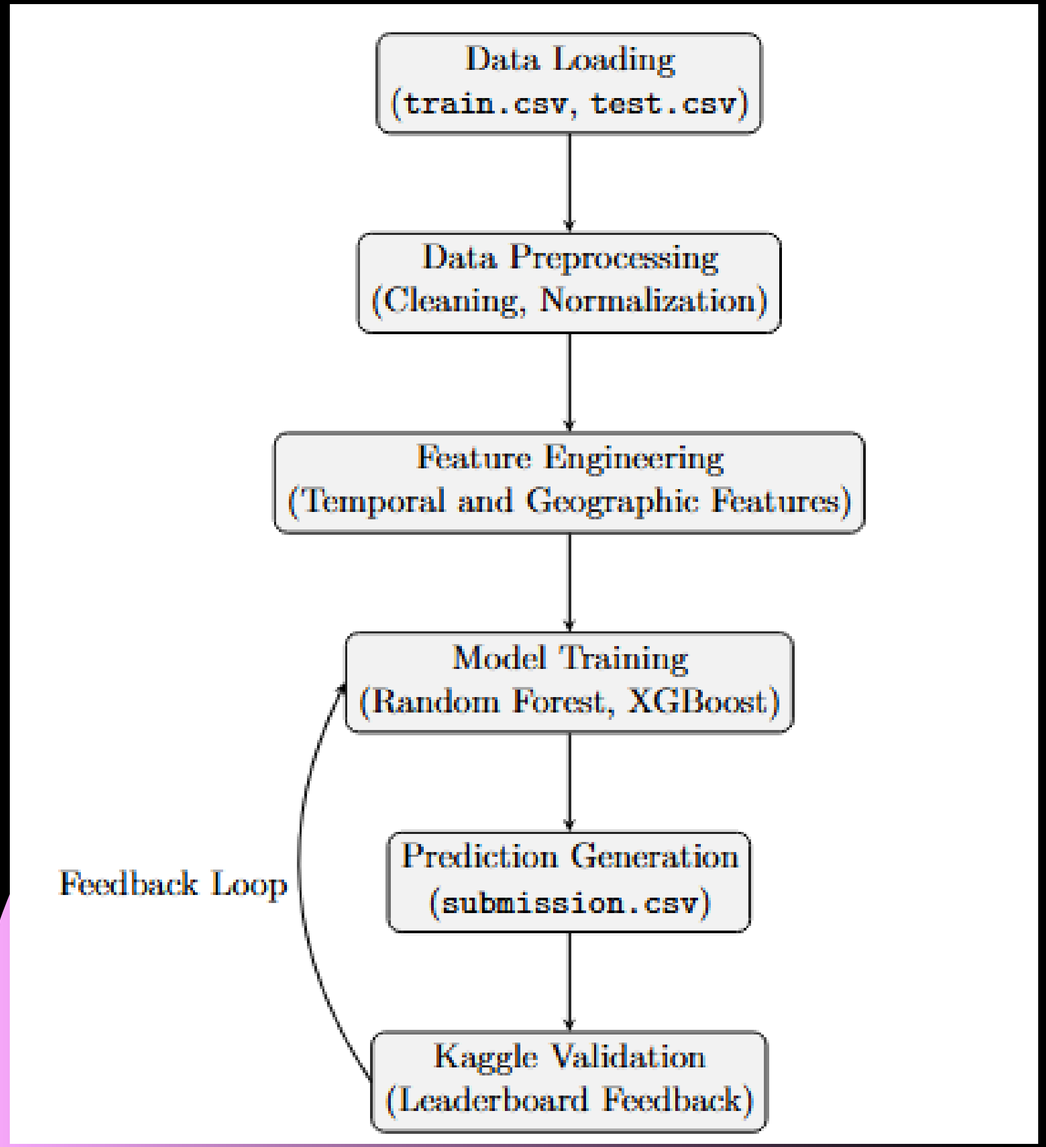
Require: train.csv, test.csv, retraining threshold δ
Ensure: submission.csv with predicted ConfirmedCases and Fatalities

1: Load train and test datasets
2: Preprocess data: clean, normalize, and engineer features
3: Train model (Random Forest or XGBoost)
4: Generate predictions for test data
5: Evaluate results using RMSLE metric
6: **if** error ¿ δ **then**
7: Retrain model with updated data
8: **end if**
9: Output submission.csv and update via Kaggle API



TECHNICAL STACK AND IMPLEMENTATI ON SKETCH

Python is used to develop the system due to its adaptability and the strength of its data science and machine learning ecosystem. This language enables effective data management, rapid prototyping, and seamless integration with analytical tools.



Thank **You**

-