Insurance
Data
Science

Boost Climate Risk Modelling with Large Language Models Data Augmentation

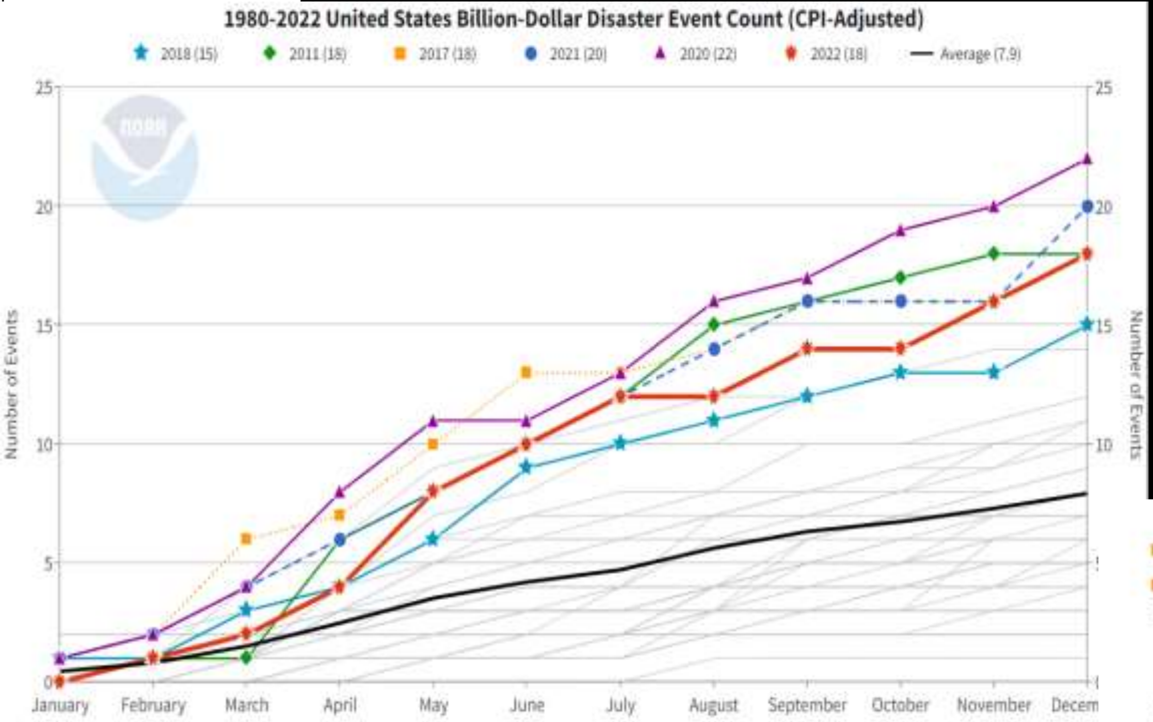
Insurance Data Science Conference, Stockholm
17th June, 2024

Claudio G. Giancaterino

TOWIN LAB
Towards Innovation Lab

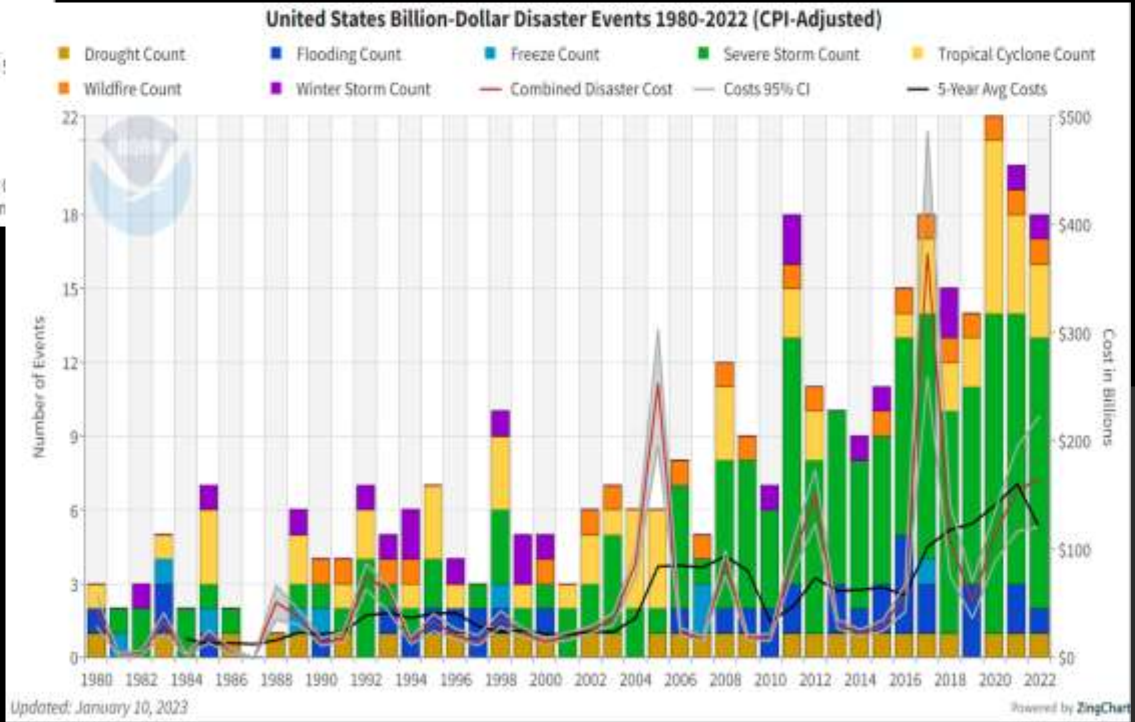
Insurance
Data
Science

Climate Change



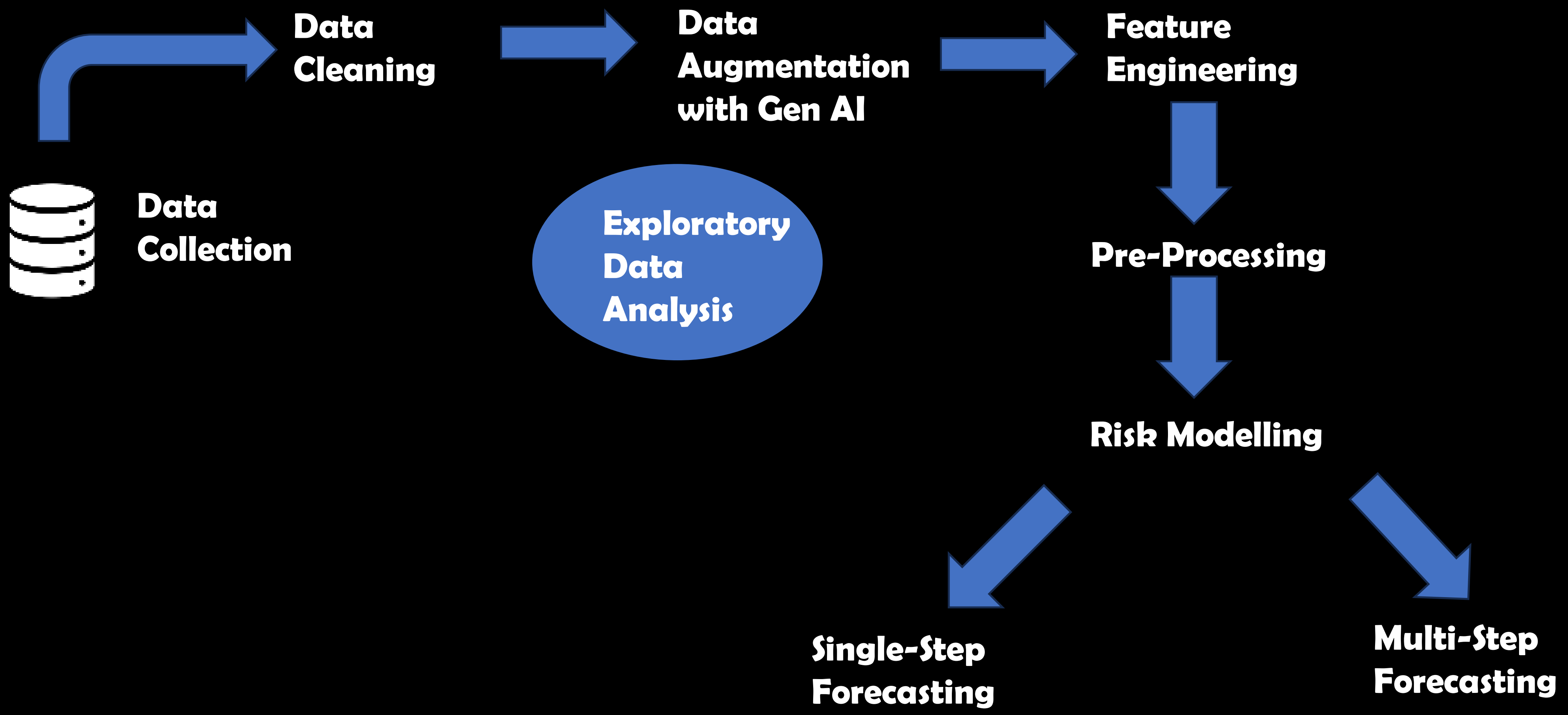
In 2023, the NOAA National Centers for Environmental Information (NCEI) released the 2022 U.S. weather and climate disasters report. Since 1980, the U.S. has experienced 341 weather and climate disasters where overall damages/costs reached or exceeded $1 billion, with a cumulative cost exceeding $2.475 trillion. Mitigating future risks requires addressing the compounding hazards driven by our changing climate.

https://www.climate.gov/news-features/blogs/beyond-data/2022-us-billion-dollar-weather-and-climate-disasters-historical



TOWIN LAB
Towards Innovation Lab

Data Flow Chart



Insurance
Data
Science

Data Collection & Data Cleaning

NCDC Storm Events Database (noaa.gov)

Data have been collected from the open source NCDC Storm Events Database.
Storm Data is provided by the National Weather Service (NWS) and contain statistics on personal injuries and damage estimates.
Storm Data covers the United States of America. The data collected began as early as 1950 through to the 2022.
The data contain a chronological listing, by state, of hurricanes, tornadoes, thunderstorms, hail, floods, drought conditions, lightning, high winds,
snow, temperature extremes and other weather phenomena.
From the website has been downloaded dataset for each year and then merged into one big dataset.
The data collected contains 51 columns included 2 text columns, and 1.794.914 rows.
The job has been done on "flood" event type, so firstly the database was realized selecting rows with the interested event type, then removing all missing values, and redundant columns or not related to the topic.
Last step involved the monetary conversion from string to numbers for the damage columns.
The final shape of the dataset: 31 columns x 38.398 records.

Flood_Event_Type

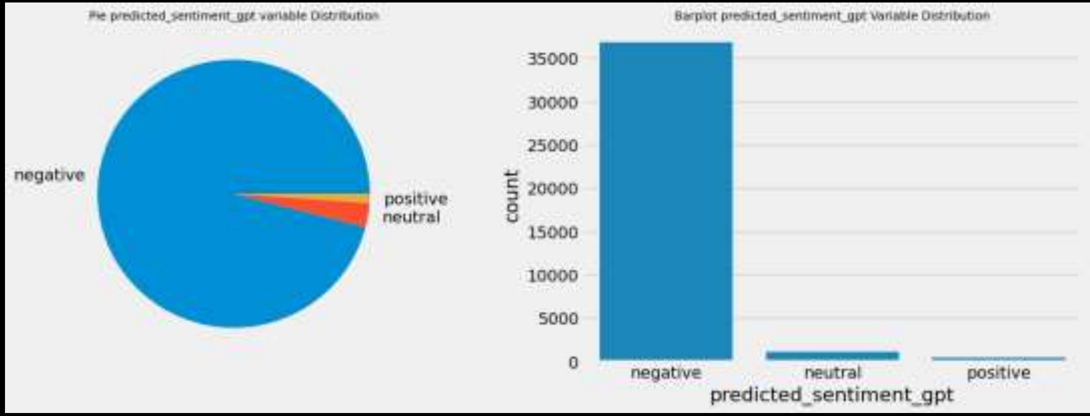
```
dftot_flood = dftot[dftot['EVENT_TYPE'].isin(['Flood'])]
```

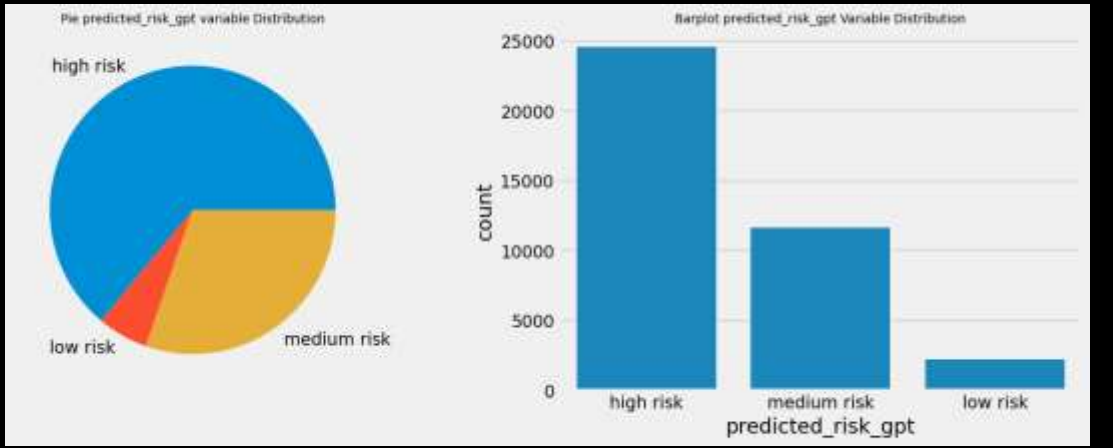
TOWIN LAB
Towards Innovation Lab

Data Augmentation with Gen AI

Insurance Data Science

Scikit-LLM is a Python library that incorporates large language models into the scikit-learn framework.
It's a tool to perform Natural Language Processing (NLP) tasks all within the Scikit-Learn pipeline.
It started integrating OpenAI models.
In this role, GPT-3.5 turbo (ChatGPT engine) has been used for data augmentation by constructing features using zero-shot text classification and text vectorization.






```
df.iloc[:,32:52].describe().T
```

	count	mean	std	min	25%	50%	75%	max
embed_episod_gpt0	38398.0	-0.016808	0.009427	-0.052020	-0.023349	-0.016708	-0.010367	0.021528
embed_episod_gpt1	38398.0	-0.006413	0.010204	-0.045055	-0.013191	-0.005958	0.000404	0.030149
embed_episod_gpt2	38398.0	0.004548	0.009586	-0.034101	-0.001928	0.004899	0.011089	0.038639
embed_episod_gpt3	38398.0	-0.008144	0.010529	-0.047241	-0.015404	-0.008313	-0.001156	0.035343
embed_episod_gpt4	38398.0	-0.005666	0.011905	-0.055183	-0.013564	-0.005558	0.002819	0.037800
embed_episod_gpt5	38398.0	0.020970	0.009892	-0.018789	0.014344	0.021122	0.027610	0.059796
embed_episod_gpt6	38398.0	-0.005247	0.010494	-0.046455	-0.012238	-0.004995	0.001793	0.039692
embed_episod_gpt7	38398.0	-0.013841	0.010543	-0.050245	-0.020813	-0.014273	-0.007203	0.032356
embed_episod_gpt8	38398.0	-0.008262	0.011402	-0.054461	-0.015653	-0.008077	-0.000625	0.032176
embed_episod_gpt9	38398.0	-0.025418	0.009086	-0.062480	-0.031091	-0.025496	-0.019398	0.010344
embed_event_gpt0	38398.0	-0.005881	0.009103	-0.043164	-0.012045	-0.005965	0.000182	0.034685
embed_event_gpt1	38398.0	-0.005816	0.010580	-0.049058	-0.012815	-0.005692	0.001314	0.037376
embed_event_gpt2	38398.0	0.002403	0.010754	-0.047314	-0.004862	0.002595	0.009719	0.044456
embed_event_gpt3	38398.0	0.001375	0.010144	-0.043979	-0.005255	0.001394	0.008207	0.039966
embed_event_gpt4	38398.0	-0.016768	0.011458	-0.058653	-0.024500	-0.016564	-0.009045	0.032566
embed_event_gpt5	38398.0	0.015199	0.011297	-0.028740	0.007385	0.015186	0.022911	0.059868
embed_event_gpt6	38398.0	-0.011737	0.011669	-0.051402	-0.019752	-0.011663	-0.003751	0.036160
embed_event_gpt7	38398.0	-0.007407	0.010022	-0.048194	-0.014242	-0.007453	-0.000847	0.030899
embed_event_gpt8	38398.0	-0.005101	0.013574	-0.052318	-0.014769	-0.004464	0.004788	0.042354
embed_event_gpt9	38398.0	-0.022895	0.009566	-0.058607	-0.029435	-0.023082	-0.016529	0.018322

TOWIN LAB
Towards Innovation Lab

Insurance
Data
Science

Feature Engineering

In this process, new features were created by extracting the year, month, day, and time from the beginning and end of each event.
The difference between dates was extrapolated in days and hours, origin and destination names were merged, categorical features were encoded, grouping less relevant classes. New target variables and the distance between the starting and ending points of the event were calculated using the haversine distance.

Target Variables
- Injuries Direct
- Deaths Direct
- Damage Property
- Injuries Indirect
- Deaths Indirect
- Damage Crops

New Target Variables
- Injuries Direct
- Deaths Direct
- Damage Property
- Whole Injuries
- Whole Deaths
- Whole Damage

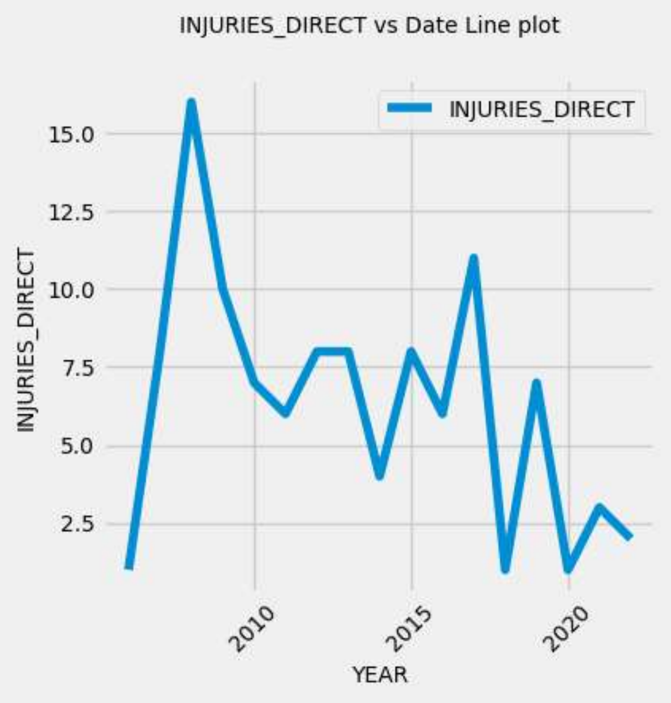
```python
# Distance of the event
def haversine_distance(row):
    # Convert latitude and longitude from degrees to radians
    lat1, lon1 = radians(row['BEGIN_LAT']), radians(row['BEGIN_LON'])
    lat2, lon2 = radians(row['END_LAT']), radians(row['END_LON'])

    # Haversine formula
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
    c = 2 * atan2(sqrt(a), sqrt(1-a))
    radius_of_earth = 6371  # Earth's radius in kilometers
    distance = radius_of_earth * c

    return distance
```

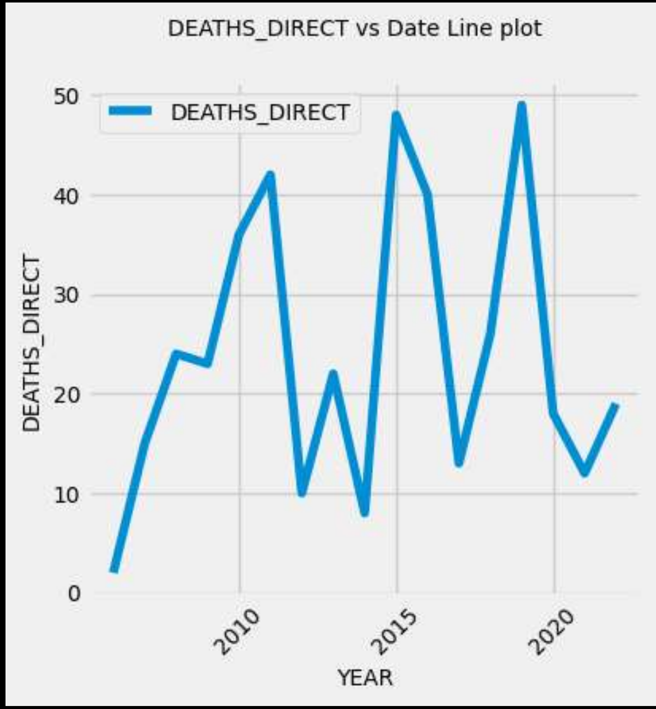
TOWIN LAB
Towards Innovation Lab

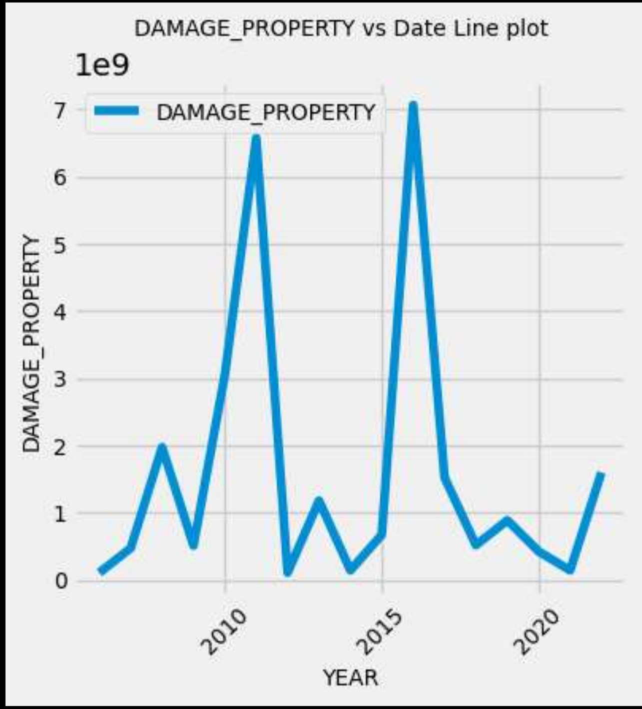
Insurance
Data
Science

Exploratory Data Analysis

**After data cleaning and data augmentation, the dataset reached the following shape:
53 columns and 38398 rows. The observations span from 2006 to 2022.
-The Injuries Direct target variable shows a peak of injuries in 2006, followed by a decreasing number in the subsequent years with a second peak in 2017. California had the highest number of injuries registered.
-The Deaths Direct target variable shows fluctuating behaviour, with peaks in mortality observed in 2011, 2015, and 2019. Kentucky, Missouri, and North Carolina had the highest number of deaths observed.
-The Damage Property target variable indicates two peaks of disasters in 2011 and 2016, with Lousiana having the highest level of damage.**







TOWIN LAB
Towards Innovation Lab

Pre-Processing

Removed outliers and large damage

Removed zero variance predictors and correlated predictors

Features scaling for linear models and neural networks

Towards Innovation Lab

Insurance
Data
Science

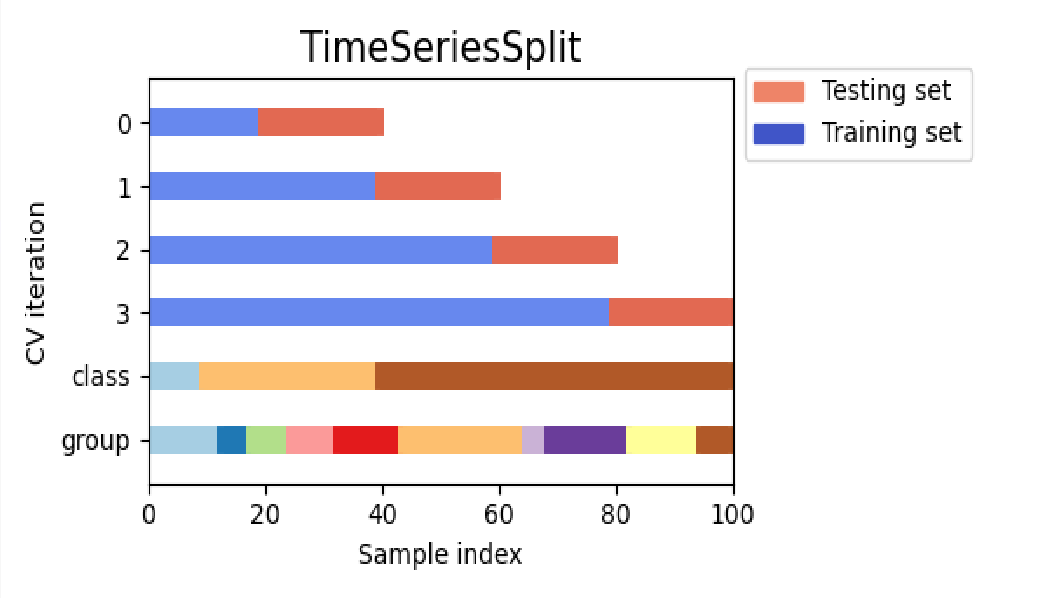
Risk Modelling

- ➢ **Naive Forecasting as benchmark**
- ➢ **Generalized Linear Models (GLM) using Tweedie distribution**
- ➢ **Gradient Boosting Machine by LightGBM**
- ➢ **Feed-Forward Deep Networks with 3 hidden layers**

To fine-tuning models have been employed a time series cross validation with optuna, an automatic hyperparameter optimization software framework, which uses bayesian optimization.

Fine-tuning

```python
def objective(trial):
    params = {
        'power': trial.suggest_loguniform('power', 1.5, 2),
        'link':'log'
    }
    np.random.seed(0)
    glm = TweedieRegressor(**params)

    tscv = TimeSeriesSplit(n_splits=5)
    mgd_scores = []

    for train_index, val_index in tscv.split(X_train_sc):
        X_tr, X_val, tr_weights_sc = X_train_sc.iloc[train_index], X_train_sc.iloc[val_index], train_weights_sc[train_index]
        y_tr, y_val = y_train.iloc[train_index], y_train.iloc[val_index]

        glm.fit(X_tr, np.maximum(y_tr,1e-12), sample_weight=tr_weights_sc)
        pred_val = glm.predict(X_val)

        mgd_score = mean_gamma_deviance(np.maximum(y_val,1e-12), pred_val)
        mgd_scores.append(mgd_score)

    return np.mean(mgd_scores)

study = optuna.create_study(direction='minimize')
study.optimize(objective, n_trials=10)

best_params = study.best_params
best_params
```



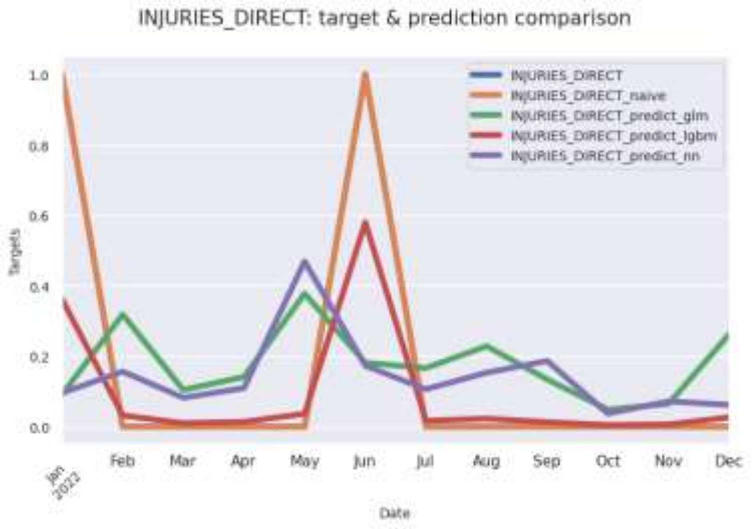
Insurance
Data
Science

Single-Step Forecasting (Predictions)

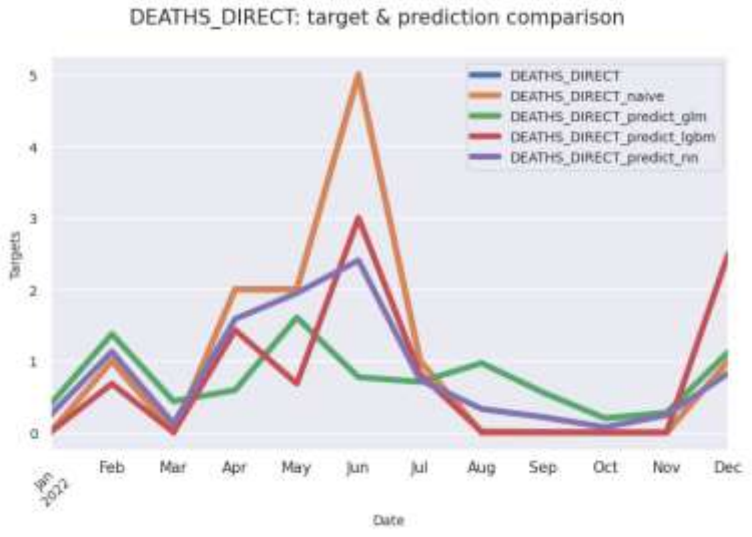
Test:
2022

Train:
2006-2021

Injuries Direct



Deaths Direct



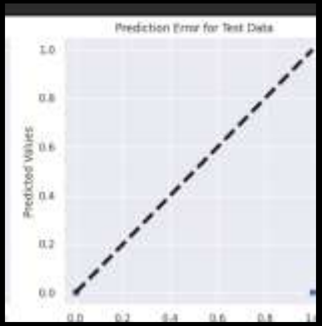
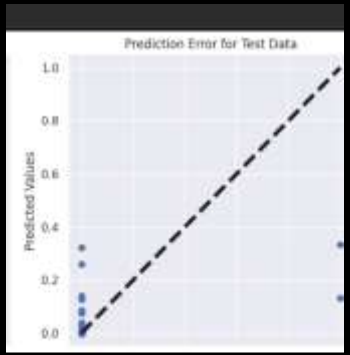
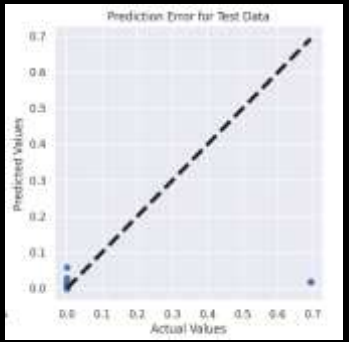
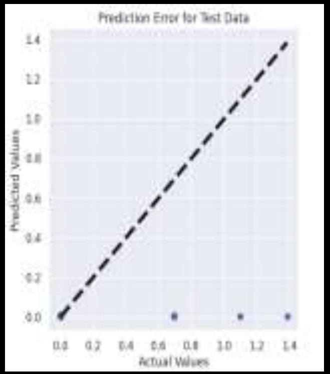
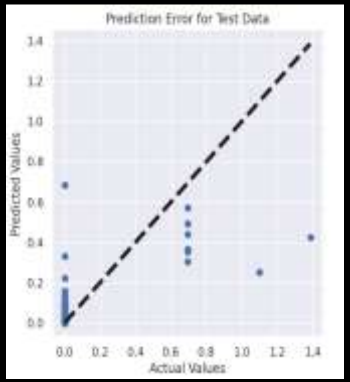
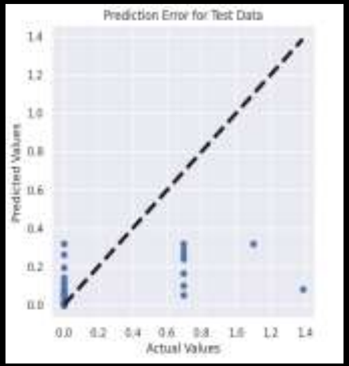
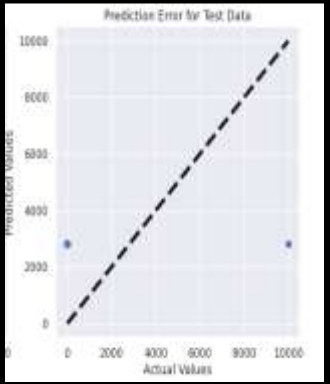
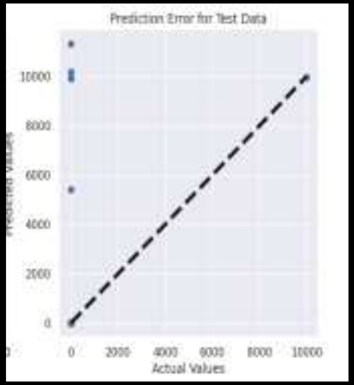
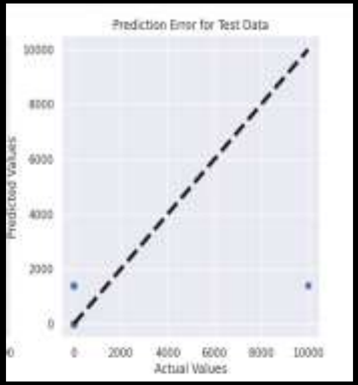
Damage Property

LightGBM is able to generalize well observations for the all outcomes.

TOWIN LAB
Towards Innovation Lab

Insurance
Data
Science

Single-Step Forecasting (Residuals on Test data)

	GLM	LightGBM	Neural Networks
Injuries Direct			
Deaths Direct			
Damage Property			

Towards Innovation Lab

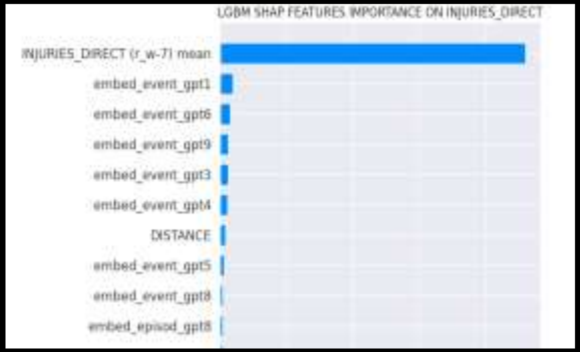
Insurance Data Science

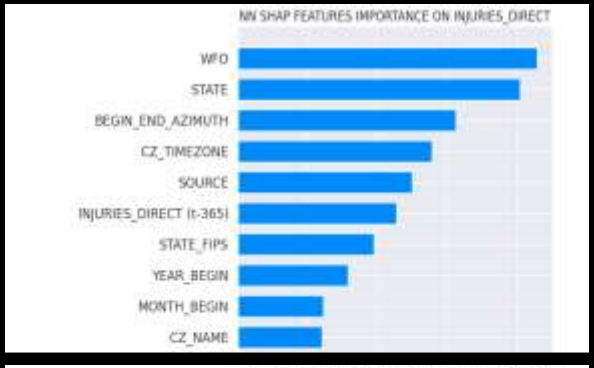
Single-Step Forecasting (Features Importance)

LightGBM is the model that exploits feature engineering and data augmentation more than the others

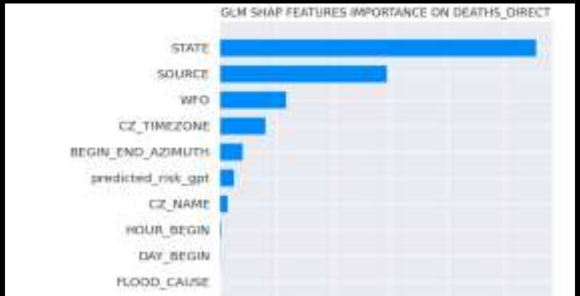
	GLM	LightGBM	Neural Networks

Injuries Direct

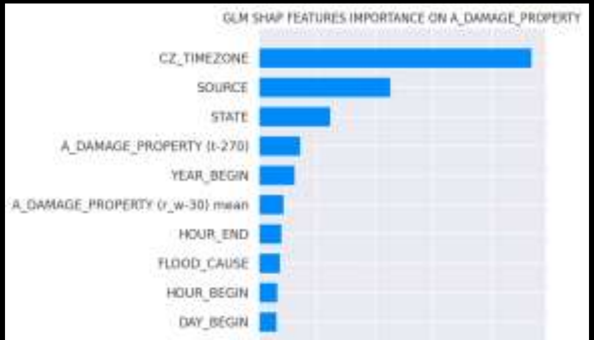


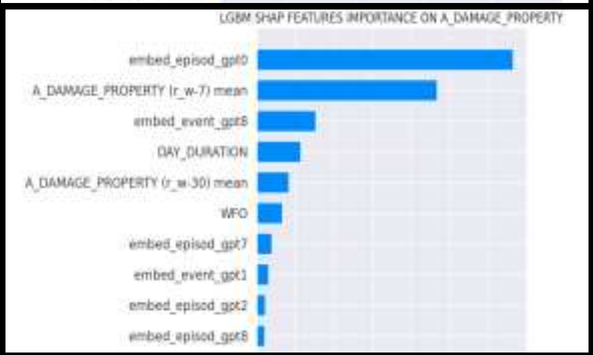


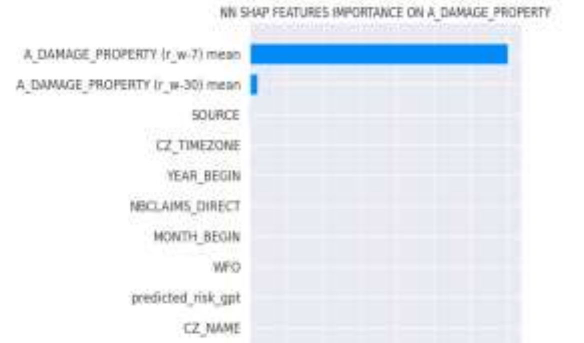
Deaths Direct



Damage Property



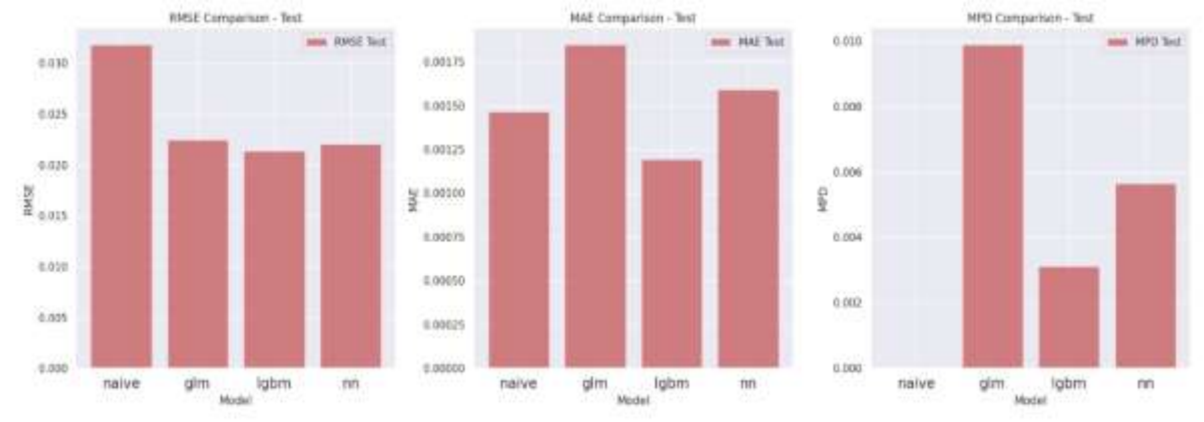




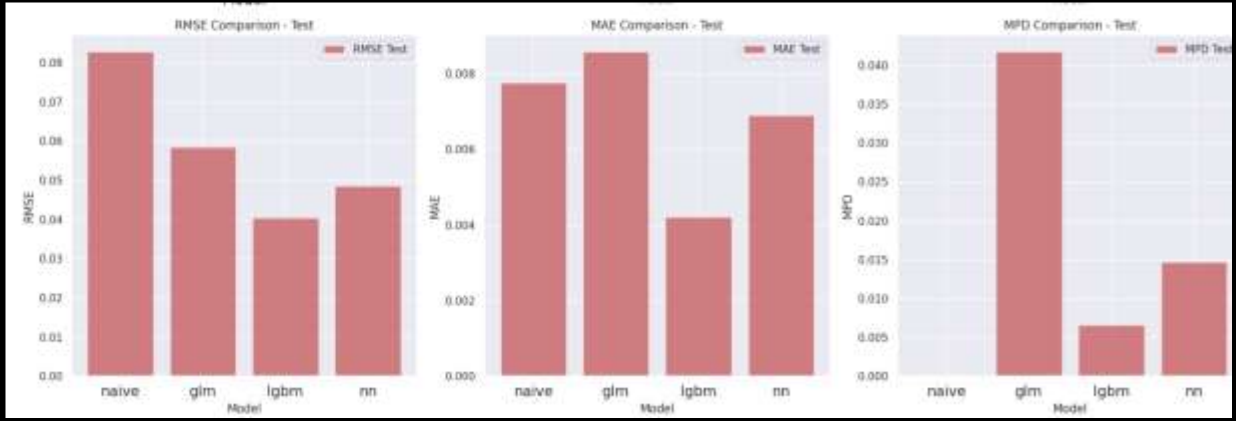
Insurance
Data
Science

Single-Step Forecasting (Performance on Test)

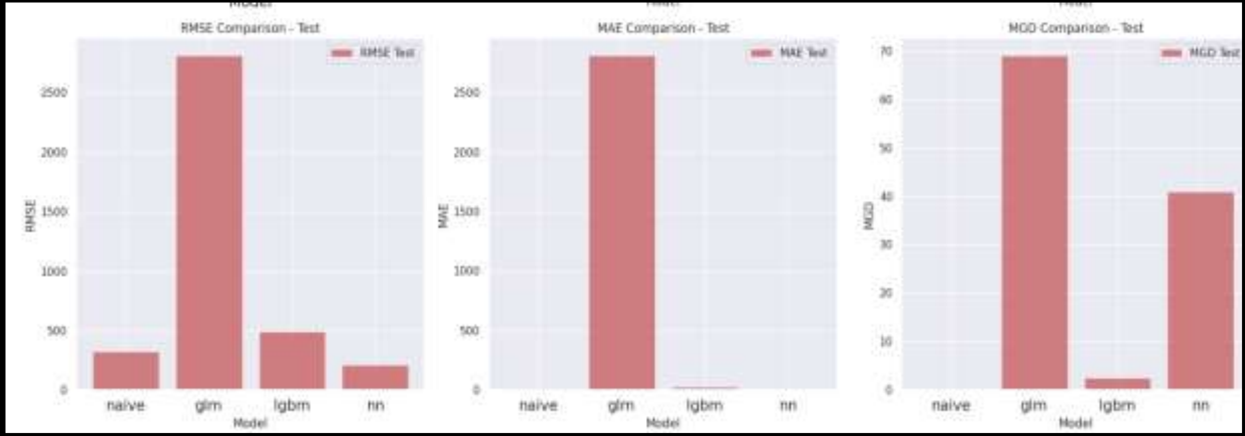
Injuries Direct



Deaths Direct



Damage Property



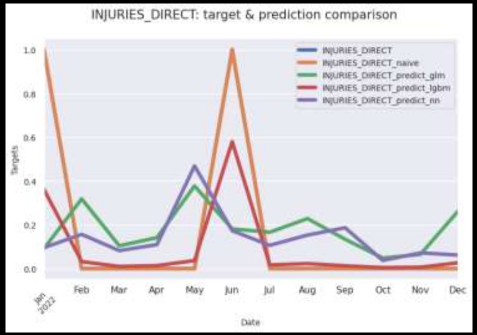
LightGBM is outstanding in performance.

TOWIN LAB
Towards Innovation Lab

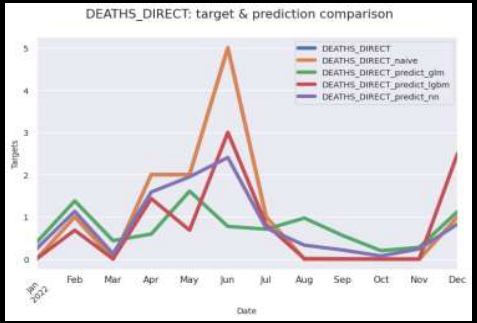
Single-Step Forecasting (augmentation vs no augmentation)

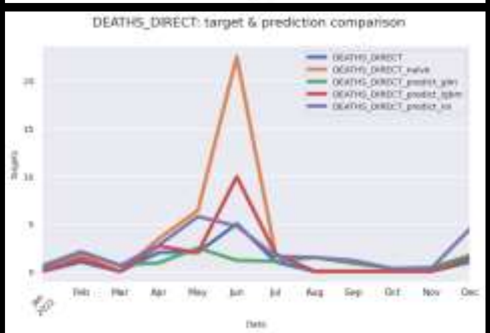
LightGBM and Neural Networks benefit from data augmentation in all predictions.

Injuries Direct



Deaths Direct



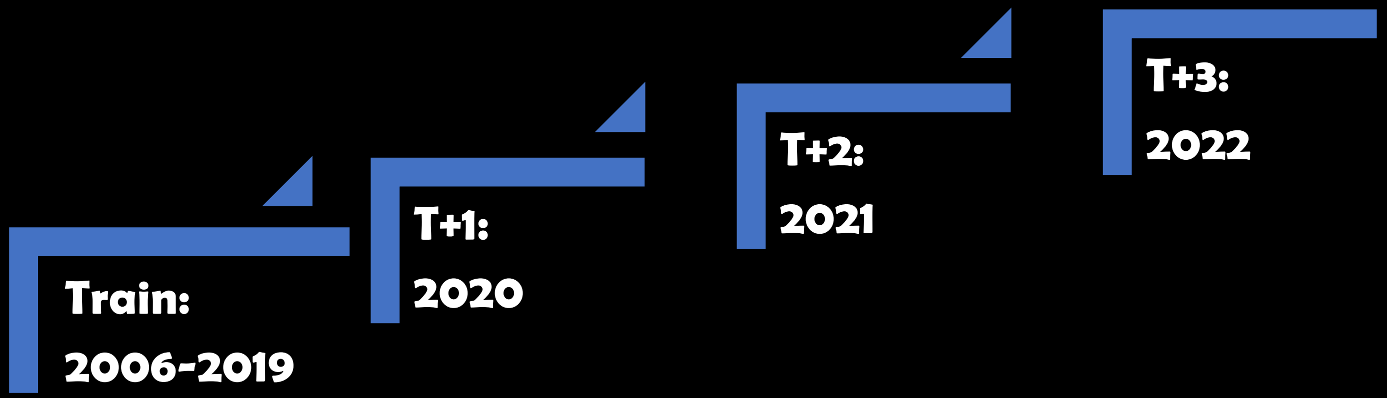


Damage Property

Towards Innovation Lab

Multi-Step Forecasting

- In single step forecasting, the goal is to predict just the next time point: t -> t+1
- In multi-step forecasting the goal is to predict the next horizon time points into the future, with h>1 and predict t+1, t+2, t+3…
- One technique is the Recursive Forecasting: train one model and use it recursively for each step of the horizon.



Insurance
Data
Science

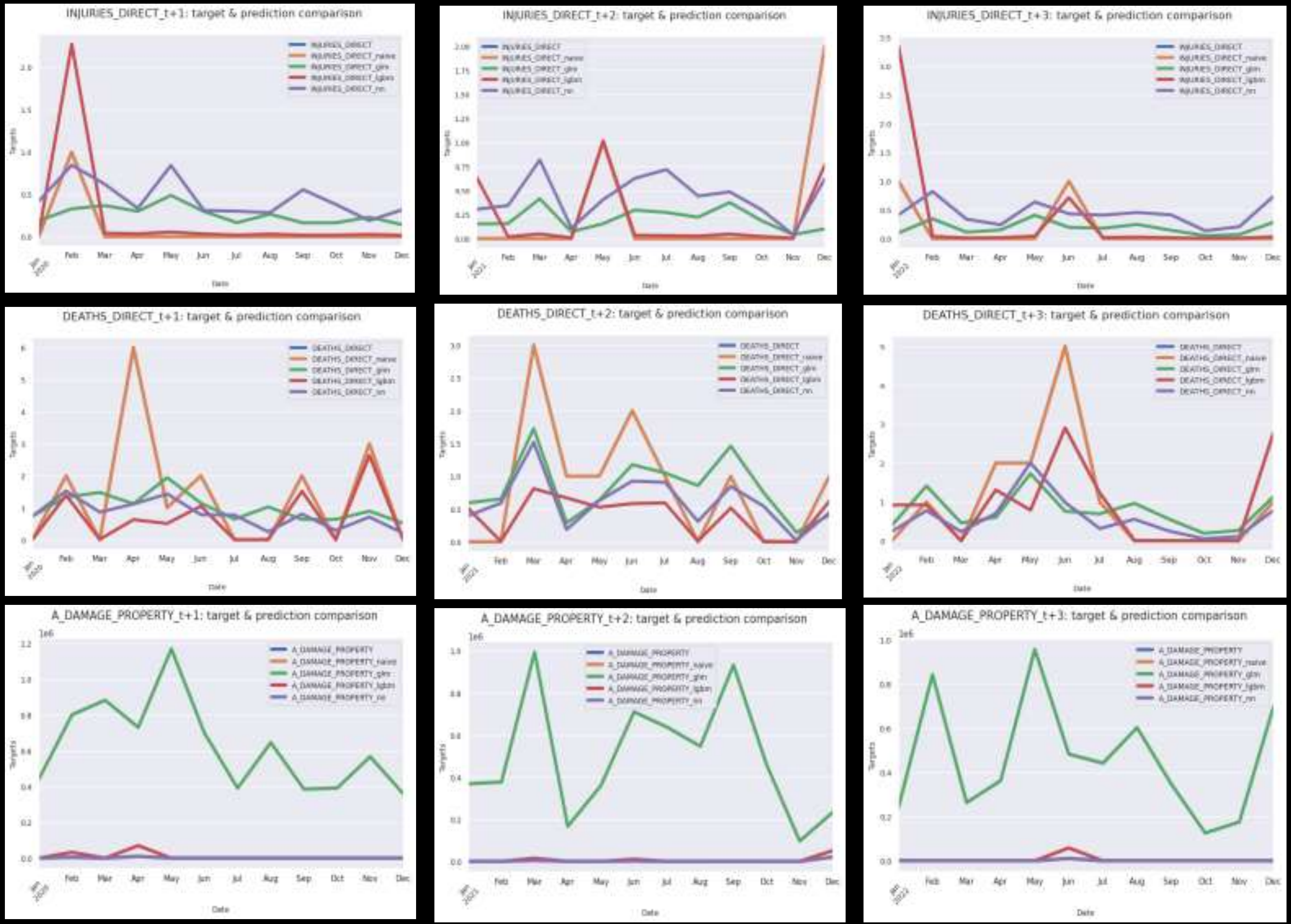
Multi-Step Forecasting (Predictions)

T+1 | T+2 | T+3

Injuries Direct

Deaths Direct

Damage Property



LightGBM generalizes well across different time series and forecasting horizons.

TOWIN LAB
Towards Innovation Lab

Conclusions

- ✓ Fighting climate risk events requires big data to improve prediction and understanding features involved in these events.
- ✓ This study showcases the potential of integrating modern Machine Learning and Generative AI to enhance climate change modelling and prediction. The results highlight the importance of feature engineering in general and the role of features extracted and generated by LLMs.
- ✓ Modern Machine Learning and Generative AI can be used to improve the mapping of high-risk zones providing more accurate quotes.

References

- P.J. Brockwell, R.A. Davis (2016). *Introduction to Time Series and Forecasting*, Springer.
- J. Birkmann, T. Welle (2015). *Assessing the risk of loss and damage: exposure, vulnerability and risk to climate-related hazards for different country classifications*, International Journal of Global Warming.
- Dineva Snezhana, (2023). *Applying Artificial Intelligence (AI) for Mitigation Climate Change Consequences of the Natural Disasters*, SSRN .
- Adam B. Smith (2022). *2022 U.S. billion-dollar weather and climate disasters in historical context*, NOAA National Centers for Environmental Information.
- Data: Index of /pub/data/swdi/stormevents/csvfiles (noaa.gov)
- Github Repository: claudio1975/Climate_Risk_Modelling_with_LLMs (github.com)

Insurance
Data
Science

?

?

?

?

Thank you

?

?

?

Keep in touch:

- Linkedin
- Newsletter
- Medium
- Website

Towards Innovation Lab