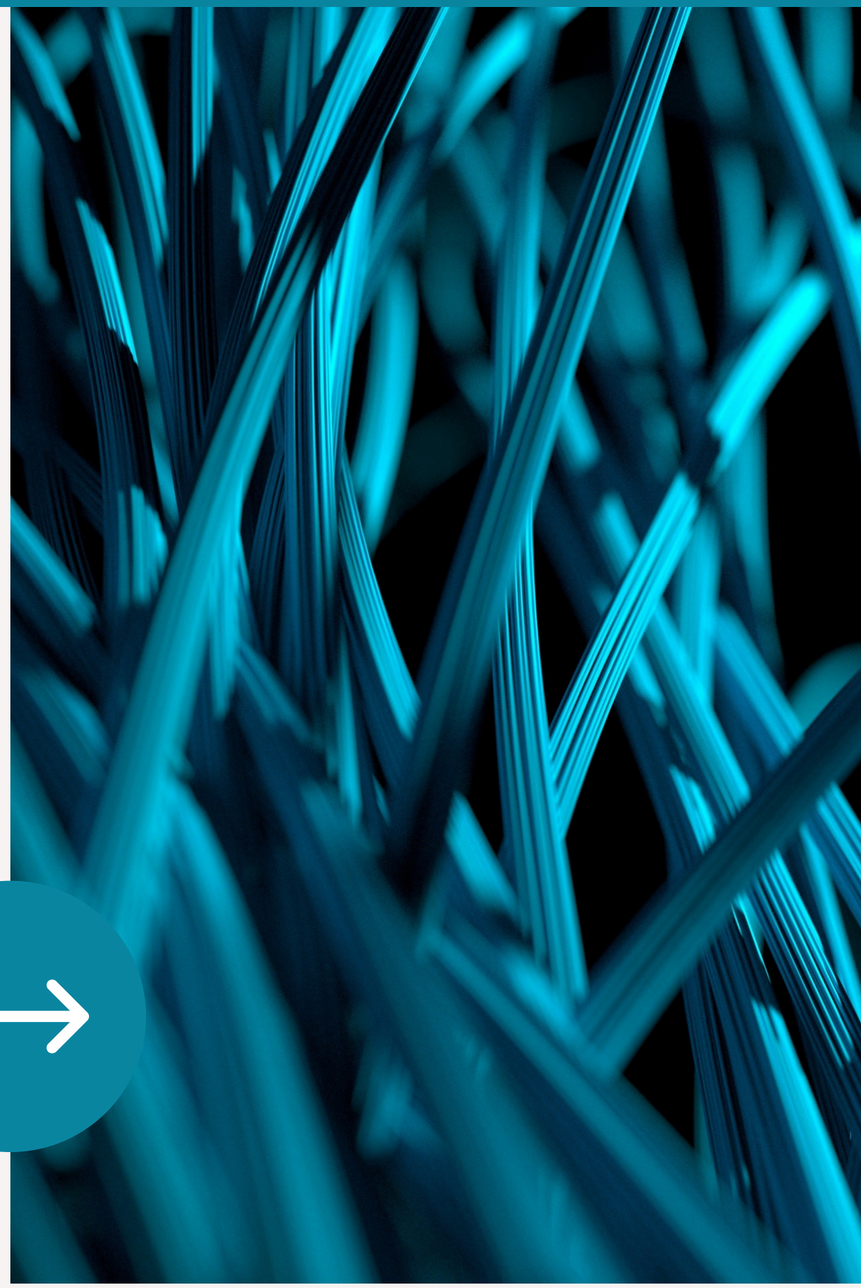
INSURANCE CLAIMS PREDICTION USING GLMS AND XGBOOST

Exploring insurance modeling techniques—from traditional GLMs to advanced machine learning with XGBoost..



INTRODUCTION

01

What is Insurance Claim Modeling?

Insurance claim modeling involves predicting the number and cost of claims made by policyholders. It's essential for pricing, reserving, and risk management.

02

Project Objective

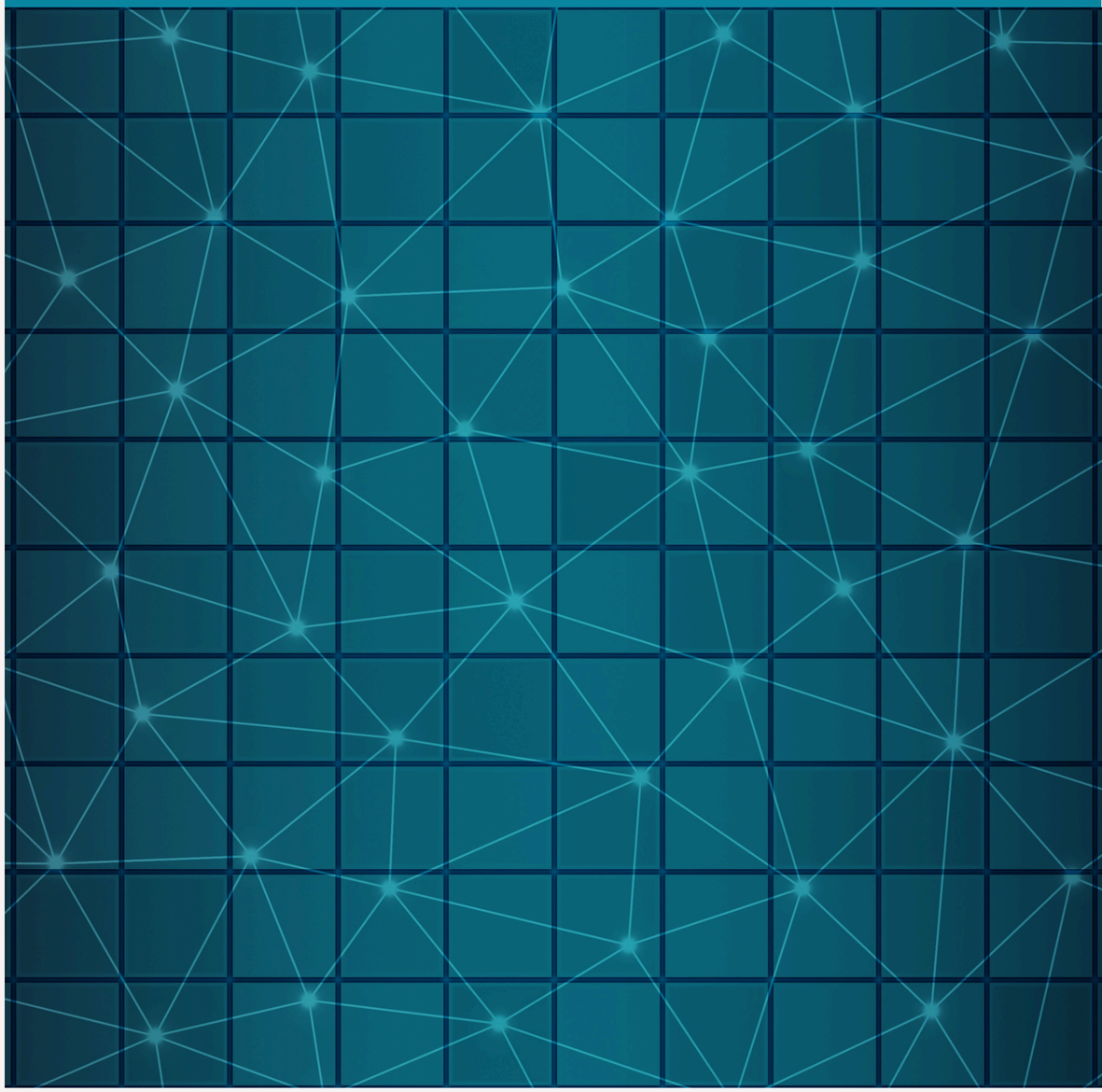
To predict total insurance claim costs we are going to be using two approachs:

- Traditional statistical models (GLMs)
- Machine learning models (XGBoost)

03

Modeling Approach: Frequency–Severity Decomposition

- Frequency: Predicting how often claims occur
- Severity: Predicting the average cost of a claim
- Total cost = Frequency × Severity



DATA PREPARATION & OVERVIEW

	PolicyID	ClaimNb	Exposure	Power	CarAge	Dr
0	1	0	0.09	g	0	
1	2	0	0.84	g	0	
2	3	0	0.52	f	2	
3	4	0	0.45	f	2	
4	5	0	0.15	g	0	

Brand	Gas	Region	Density	AvgClaimAmount
Korean	Diesel	Monastir	76	0.0
Korean	Diesel	Monastir	76	0.0
Korean	Regular	Bizerte	3003	0.0
Korean	Regular	Bizerte	3003	0.0
Korean	Diesel	Nabeul	60	0.0

Dataset :

French Motor Third-Party Liability Dataset
Provided by the CASD, this dataset is widely used for insurance modeling research.

Preprocessing Step:

The two datasets were merged based on common identifiers to allow frequency–severity decomposition and cost modeling.

Structure :

The data is split into two parts:

- Frequency Dataset: Contains policy and exposure information, including the number of claims per policy.
- Severity Dataset: Contains details about the cost of individual claims.

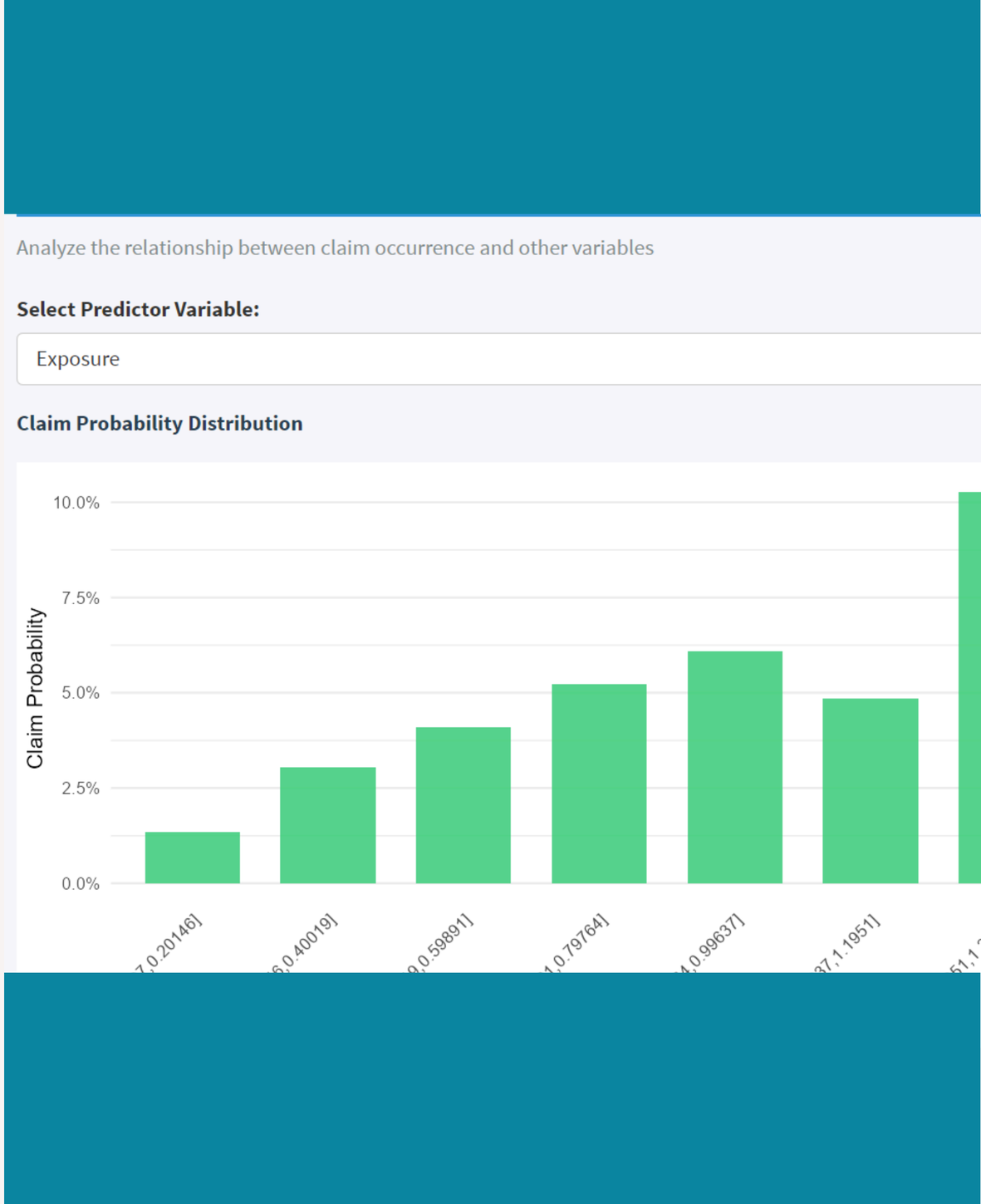
EXPLORATORY DATA ANALYSIS (EDA)

Univariate and Bivariate Analysis

- Examined key variables such as vehicle age, driver age, , and vehicle power.
- Analyzed the distribution and influence of each feature on claim outcomes.
- Investigated outliers using box plots, and assessed the validity of unusual values with external domain knowledge (e.g., car prices and city population density).

Spatial Analysis

- Visualized geographic distribution of claims across different regions in Tunisia.



MODEL EVALUATION METRICS

For evaluating GLMs and machine learning models, the following metrics were used:

- RMSE ,MAE and MAPE:

- Deviance:

  Evaluates the goodness-of-fit for GLMs; lower values indicate better fit.

- Gini Coefficient:

  Assesses the discriminatory power of the model—how well it ranks risks or predicts claim likelihood.

FREQUENCY MODELING WITH GLM

Distribution Choice

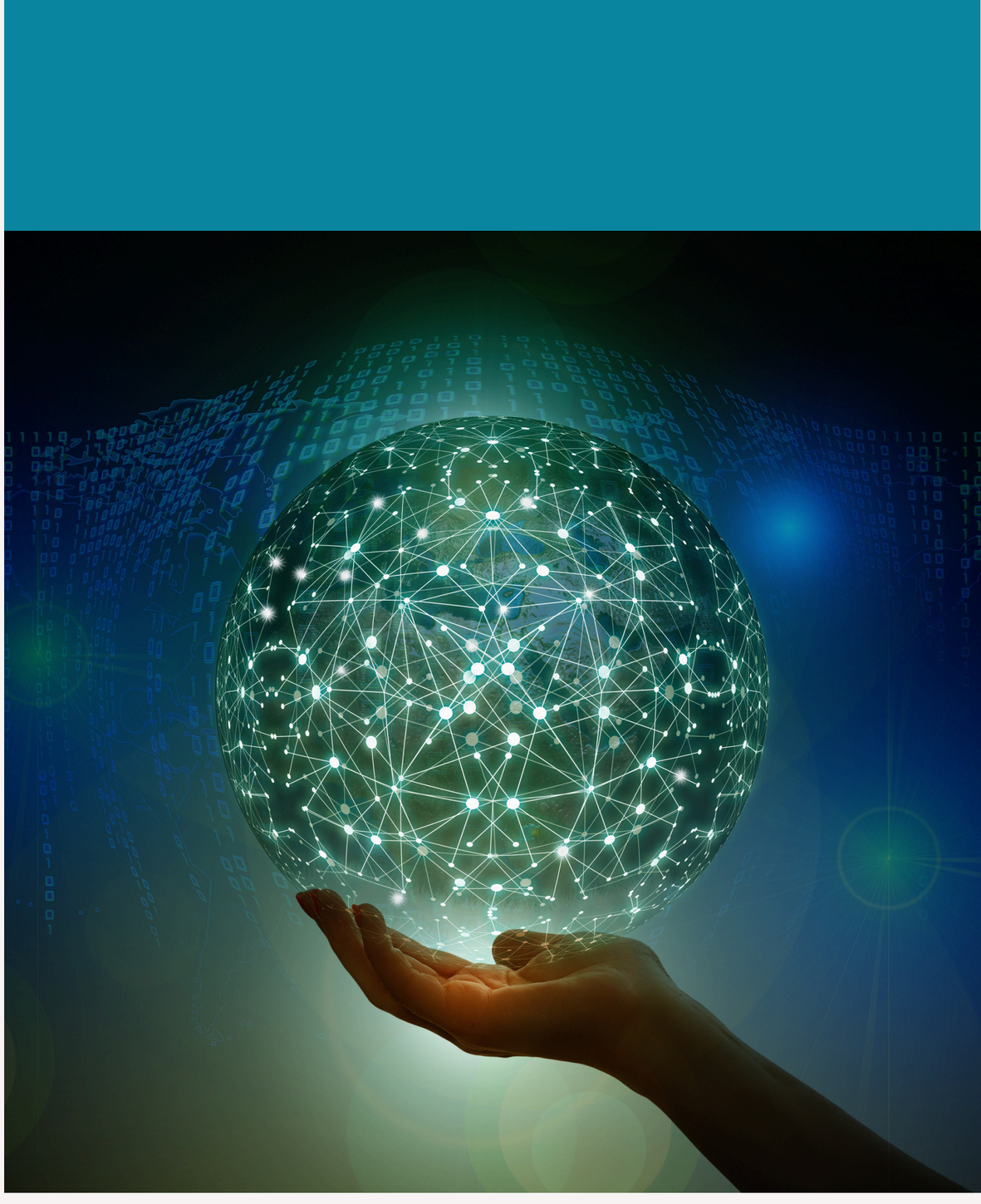
- Started with a Poisson model—commonly used for count data.
- Switched to Negative Binomial (NB) to address overdispersion (variance > mean).

Model Fitting & Performance

- Fitted GLMs using exposure as an offset.
- Evaluated models using RMSE, Deviance, and Gini coefficient.
- NB model provided a better fit than Poisson based on these metrics.

$$\mathbb{E}[N_i] = \mu_i = \exp(\mathbf{x}_i^\top \boldsymbol{\beta} + \log(\text{Exposure}_i))$$

$$\mu_i = \text{Exposure}_i \cdot \underbrace{\exp(\mathbf{x}_i^\top \boldsymbol{\beta})}_{\text{Predicted Claim Rate}}$$



FREQUENCY MODELING WITH GLM

Handling Categorical Variables

- Applied appropriate encoding to categorical features (e.g., region, vehicle type).
- Grouped or dropped modalities with low frequency or non-significant effects.

Zero-Inflated Negative Binomial (ZINB)

- Tested ZINB model to further account for excess zeros in the data.
- Compared ZINB performance with standard NB—found limited improvement in this case.

	coef	std err	z	P>|z|	[0.025	0.975]
	-2.0886	0.052	-40.274	0.000	-2.190	-1.987
	0.0661	0.032	2.069	0.039	0.003	0.129
	0.0872	0.031	2.790	0.005	0.026	0.149
	0.0609	0.031	1.964	0.049	0.000	0.122
	0.0828	0.045	1.852	0.064	-0.005	0.170
	0.2291	0.049	4.706	0.000	0.134	0.324
	0.1592	0.050	3.167	0.002	0.061	0.258
	0.2506	0.064	3.926	0.000	0.126	0.376
	0.1435	0.094	1.526	0.127	-0.041	0.328
	0.1751	0.136	1.289	0.197	-0.091	0.441
	0.2685	0.157	1.712	0.087	-0.039	0.576
	0.2332	0.156	1.497	0.134	-0.072	0.539
Nissan) or Korean]	-0.2619	0.052	-5.039	0.000	-0.364	-0.160
r or BMW]	-0.0067	0.061	-0.110	0.912	-0.125	0.112
ors or Ford]	0.0473	0.051	0.929	0.353	-0.053	0.147
r Citroen]	-0.0759	0.045	-1.704	0.088	-0.163	0.011
Skoda or Seat]	0.0120	0.052	0.229	0.819	-0.091	0.114
	-0.1294	0.072	-1.799	0.072	-0.270	0.012
	-0.1422	0.020	-7.244	0.000	-0.181	-0.104
	-0.1124	0.085	-1.317	0.188	-0.280	0.055
	0.2185	0.088	2.470	0.013	0.045	0.392
	0.0038	0.055	0.069	0.945	-0.105	0.112

	coef	std err	z	P>|z|	[0.025	0.97
	-2.0940	0.052	-40.560	0.000	-2.195	-1.9
	0.0762	0.032	2.400	0.016	0.014	0.1
	0.0853	0.031	2.733	0.006	0.024	0.1
	0.0626	0.031	2.025	0.043	0.002	0.1
	0.1166	0.040	2.908	0.004	0.038	0.1
	0.2326	0.048	4.828	0.000	0.138	0.3
	0.1759	0.050	3.537	0.000	0.078	0.2
	0.2573	0.063	4.065	0.000	0.133	0.3
	-0.2648	0.051	-5.168	0.000	-0.365	-0.1
Seat , other]	-0.0468	0.044	-1.065	0.287	-0.133	0.0
	-0.1400	0.019	-7.196	0.000	-0.178	-0.1
	-0.0449	0.039	-1.139	0.255	-0.122	0.0
	0.2236	0.088	2.531	0.011	0.050	0.3
	-0.1293	0.041	-3.190	0.001	-0.209	-0.0
	-0.1684	0.066	-2.542	0.011	-0.298	-0.0
	0.3579	0.021	16.790	0.000	0.316	0.4
	0.3266	0.050	6.506	0.000	0.228	0.4
	0.4667	0.100	4.651	0.000	0.270	0.6
	0.4344	0.049	8.889	0.000	0.339	0.5

```python
def predire_cout_total(df, glm_freq, glm_sev):


    # Prédiction du nombre de sinistres (fréquence)
    df["freq_pred"] = glm_freq.predict(df)

    # Prédiction du coût moyen des sinistres (sévérité) uniquement pou
    df["sev_pred"] = glm_sev.predict(df)

    # Calcul du coût total = fréquence prédite × sévérité prédite
    df["cout_total_pred"] = df["freq_pred"] * df["sev_pred"]

    return df["cout_total_pred"]
```

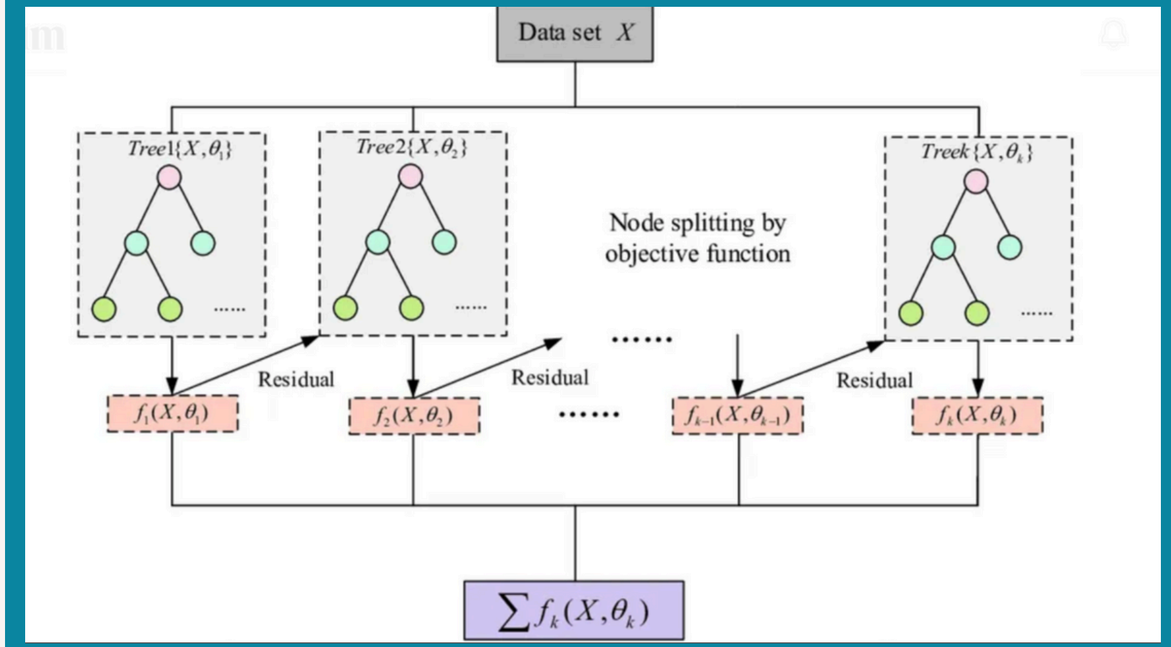
SEVERITY MODELING WITH GLM

Choice of Distribution
Used the Gamma distribution, which is suitable for modeling continuous, positive like claim amounts.

COST MODELING

- Multiplying frequency and severity models
- Evaluating final cost predictions

FREQUENCY NON-LINEAR MODELING WITH XGBOOST

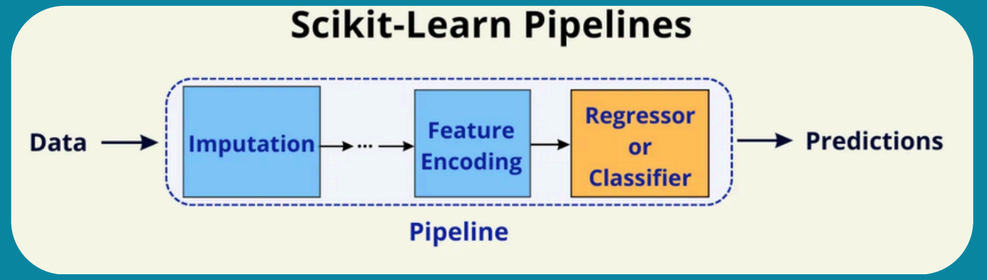


Motivation: Why XGBoost?

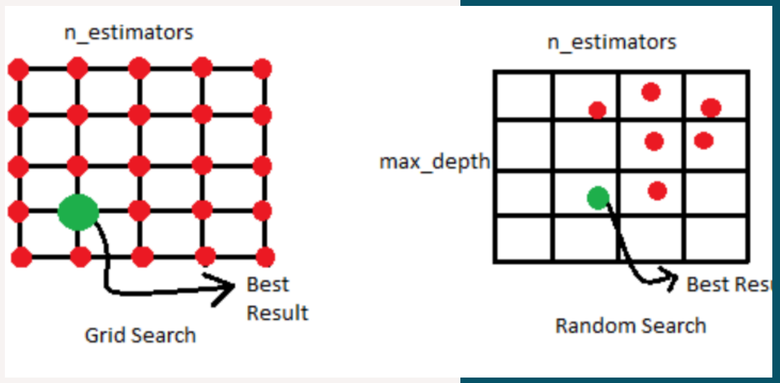
- GLMs assume linear, additive effects, which can limit performance.
- XGBoost captures non-linearities, feature interactions, and complex relationships without manual transformation.
- Well-suited for the variability and complexity in insurance data

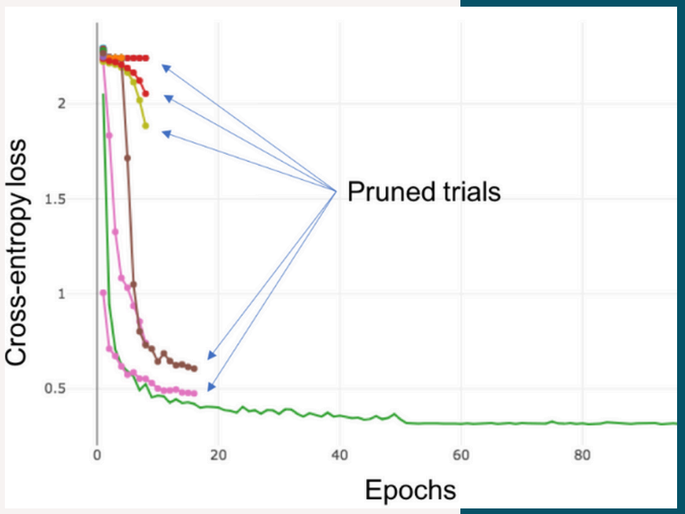
Model Training

- Trained on the merged dataset to predict the claim rate wich is claimnb/expousre
- Built the model using a Scikit-learn pipeline, which streamlined preprocessing and model integration.
- This setup also made it easier to deploy the model in a Streamlit interface for interactive use.



HYPERPARAMETER TUNING: OPTUNA VS. RANDOM SEARCH





Random Search:

- Randomly sampled combinations of hyperparameters.
- Easy to implement but potentially inefficient — may miss optimal zones.
- Used for baseline tuning performance.

Optuna (Bayesian Optimization

- Automated, intelligent search based on past results (Tree-structured Parzen Estimator).
- Focused the search on promising regions of the hyperparameter space.
- Faster convergence to better performance compared to Random Search.

Outcome:

- Optuna outperformed Random Search in both RMSE and Gini coefficient, with random search taking less time but the two methods took the same number of trials

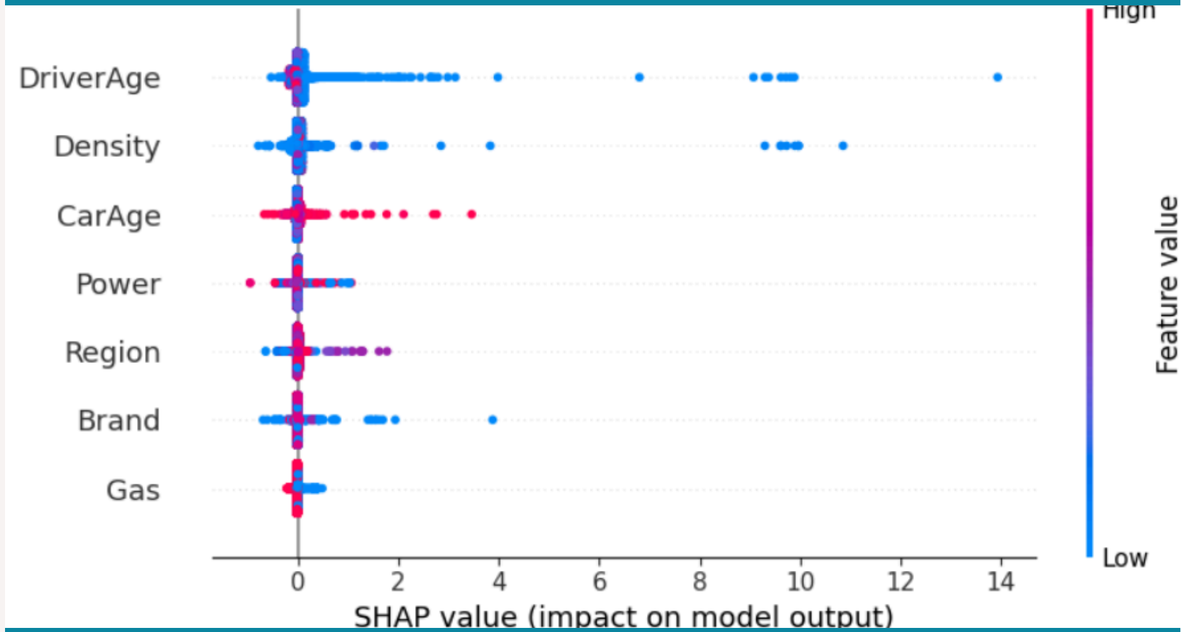
Parameters Tuned:

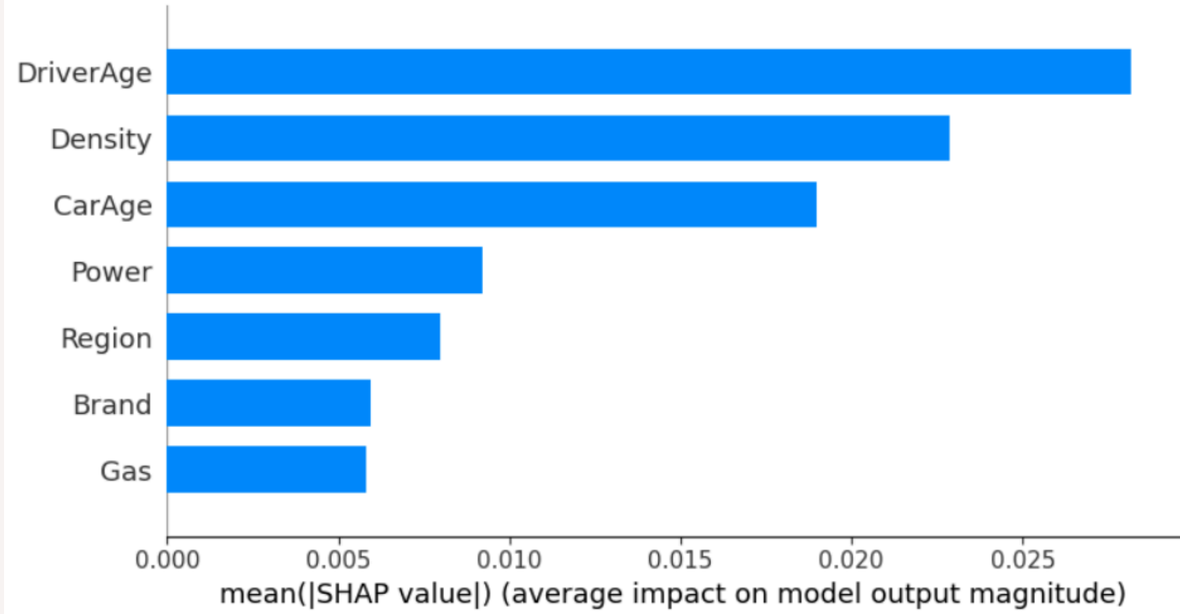
- learning_rate, max_depth, n_estimators, subsample, colsample_bytree, gamma, reg_alpha, reg_lambda

FEATURE IMPORTANCE WITH SHAP

Why SHAP? :

- SHAP (SHapley Additive exPlanations) provides a consistent, game-theoretic approach to interpret model outputs.
- Quantifies the contribution of each feature to each individual prediction.

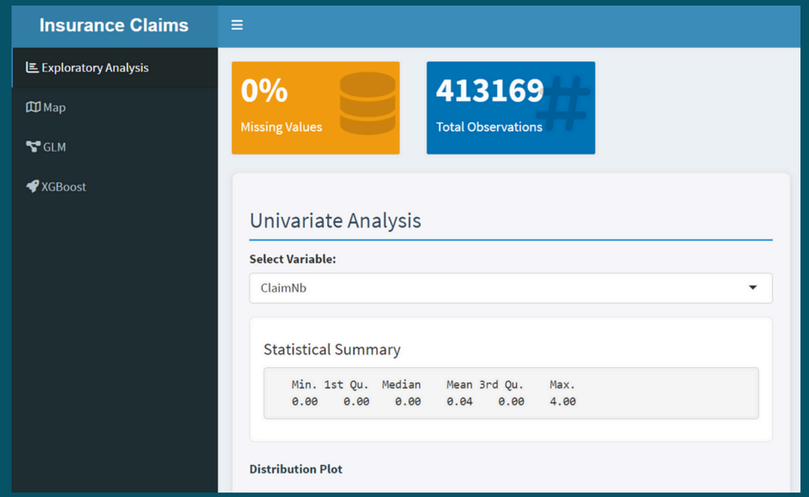




INTERACTIVE DASHBOARDS

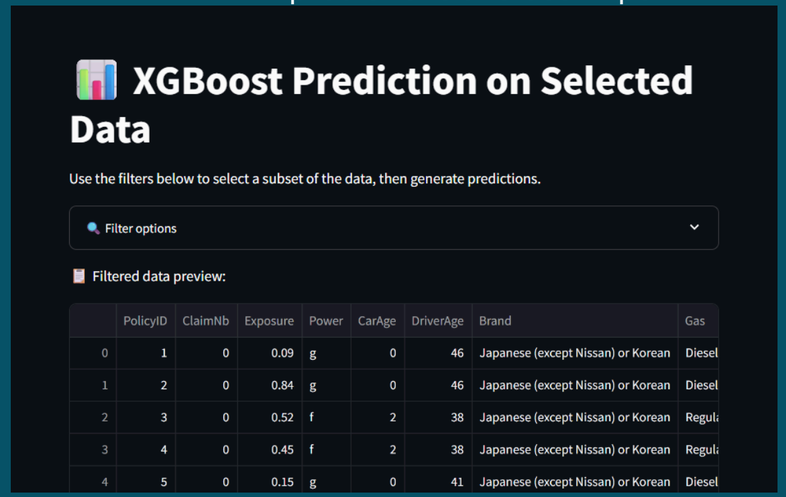
RShiny Dashboard :

- Designed for Exploratory Data Analysis (EDA) and model performance visualization.
- Included:
  - Interactive plots for feature distributions and spatial analysis
  - Model evaluation metrics (RMSE, Deviance, Gini)
  - Residual analysis to assess model fit and detect patterns



Streamlit Interface:

- Developed a Streamlit app in Python to deploy the XGBoost model.
- Integrated using a Scikit-learn pipeline for consistent preprocessing and prediction.
- Hosted live on Hugging Face Spaces, allowing real-time predictions and user input.



	PolicyID	ClaimNb	Exposure	Power	CarAge	DriverAge	Brand	Gas
0	1	0	0.09	g	0	46	Japanese (except Nissan) or Korean	Diesel
1	2	0	0.84	g	0	46	Japanese (except Nissan) or Korean	Diesel
2	3	0	0.52	f	2	38	Japanese (except Nissan) or Korean	Regul
3	4	0	0.45	f	2	38	Japanese (except Nissan) or Korean	Regul
4	5	0	0.15	g	0	41	Japanese (except Nissan) or Korean	Diesel

CONCLUSION

- Developed a complete pipeline for insurance claim cost prediction, from data cleaning and exploratory analysis to model deployment.
- Used the frequency–severity decomposition with GLMs to model claim behavior in a structured and interpretable way.
- Applied XGBoost to capture complex, non-linear patterns and improve predictive performance beyond traditional models.
- Implemented robust hyperparameter tuning with Optuna, and used SHAP values for transparent model interpretation.
- Built interactive tools with R Shiny (for EDA and diagnostics) and Streamlit (for real-time prediction), enhancing accessibility and usability.