AI in Industry Project – City of London accidents Causal Analysis

Federico Faccioli, Aaron Stephan Salazar Jaramillo, Mohammed Oubia

The dataset

- We decided to analyze a dataset including accidents in the Greater City of London (year 2019)
- Dataset downloaded from TfL – Transport for London website https://api.tfl.gov.uk
- Every entry represents an accident, for each accident we have the following features: id, Lat, Lon, Location, Date, Severity, Borough, Casualties, Vehicles

	$type	id	lat	lon	location	date	severity	borough	casualties	vehicles
0	Tfl.Api.Presentation.Entities.AccidentStats.Ac...	345979	51.570865	-0.231959	On Edgware Road Near The Junction With north C...	2019-01-04T21:22:00Z	Slight	Barnet	[{'$type': 'Tfl.Api.Presentation.Entities.Acci...	[{'$type': 'Tfl.Api.Presentation.Entities.Acci...
1	Tfl.Api.Presentation.Entities.AccidentStats.Ac...	345980	51.603859	-0.187240	On Willow Way Near The Junction With Long Lane	2019-01-04T23:33:00Z	Slight	Barnet	[{'$type': 'Tfl.Api.Presentation.Entities.Acci...	[{'$type': 'Tfl.Api.Presentation.Entities.Acci...
2	Tfl.Api.Presentation.Entities.AccidentStats.Ac...	345981	51.512198	-0.153122	On north Audley Street Near The Junction With ...	2019-01-04T22:15:00Z	Slight	City of Westminster	[{'$type': 'Tfl.Api.Presentation.Entities.Acci...	[{'$type': 'Tfl.Api.Presentation.Entities.Acci...
3	Tfl.Api.Presentation.Entities.AccidentStats.Ac...	345982	51.431480	-0.016083	On Bromley Road Near The Junction With Daneswo...	2019-01-04T18:00:00Z	Slight	Lewisham	[{'$type': 'Tfl.Api.Presentation.Entities.Acci...	[{'$type': 'Tfl.Api.Presentation.Entities.Acci...
4	Tfl.Api.Presentation.Entities.AccidentStats.Ac...	345983	51.473487	0.145202	On Belmont Road Near The Junction With Bedonwe...	2019-01-04T20:45:00Z	Slight	Bexley	[{'$type': 'Tfl.Api.Presentation.Entities.Acci...	[{'$type': 'Tfl.Api.Presentation.Entities.Acci...

Goal of the project

- Analyzing what are the causes that determine the severity of an accident
- "Severity" becomes our **target variable**
- Chosen approach:
    1. Preprocess the data
    2. Analyzing features
    3. Finding a good classification model
    4. Analyzing feature importance
    5. SHAP analysis
    6. EXTRA: Project work (by Mohammed Oubia, Aaron Salazar)

Preprocessing

- To obtain better results, we need to preprocess the dataset a bit:
  - Dropping unused features (Location)
  - Extracting useful features (Time_of_Day, Month)
  - Mapping some features to cardinal indicator (Severity, Day_of_the_week, Month)
  - Mapping some features to One-Hot-Encoding (Boroughs, Time of Day)
  - Reorganizing Vehicles and Casualties
- We will only be focusing on the causes of the accidents, dropping the features relating to the aftermath of the accidents (Vehicles involved and Casualties)

	id	lat	lon	severity	borough_Barking and Dagenham	borough_Barnet	borough_Bexley	borough_Brent	borough_Bromley	borough_Camden	...	Car	Heavy_Vehicles
0	345906	51.511963	-0.028211	0	0	0	0	0	0	0	...	2	0
1	345907	51.371636	-0.117621	0	0	0	0	0	0	0	...	2	0
2	345908	51.514951	-0.072747	0	0	0	0	0	0	0	...	0	0
3	345909	51.519173	-0.262356	0	0	0	0	0	0	0	...	1	0
4	345910	51.565743	-0.136308	0	0	0	0	0	0	0	...	1	0

Motorcycle	Other	Pedalcycle	casualty_age_0-23	casualty_age_24-30	casualty_age_31-38	casualty_age_39-50	casualty_age_50+
0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0
1	0	1	0	1	0	0	0
0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0

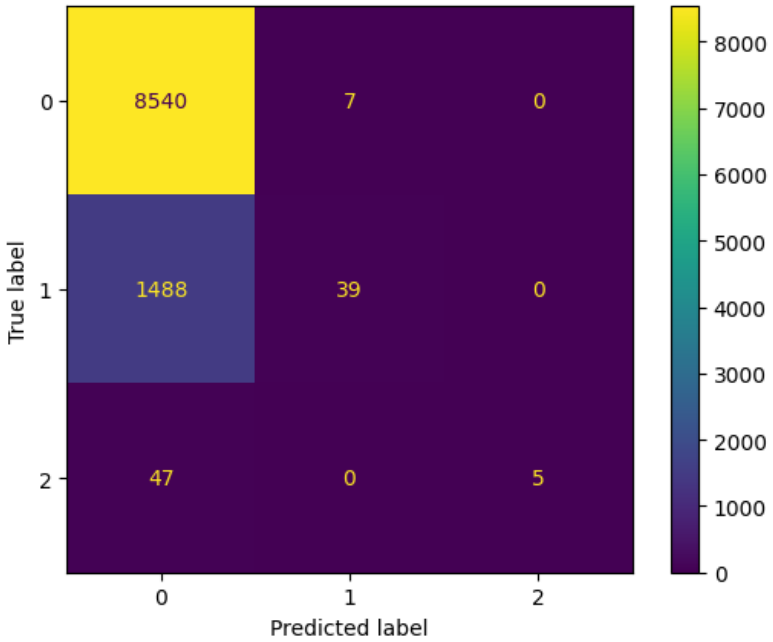
The model: discarded approaches

- CatBoost
  - No manual encoding of variables and can handle categorical values. For example, time_of_day and boroughs.
  - Built in feature importance to analyze how the model handles prediction.
  - Demonstrated too much bias towards a single class and opened the door to using balancing approaches.
- LGBM
  - Similar to CatBoost, provides memory efficiency, and is looked at for large datasets as a good option.
  - Uses optimal split for categorical features: Can learn inside of categories when to match them together
  - Demonstrated being an over complex solution to our problem and did not provide good results.

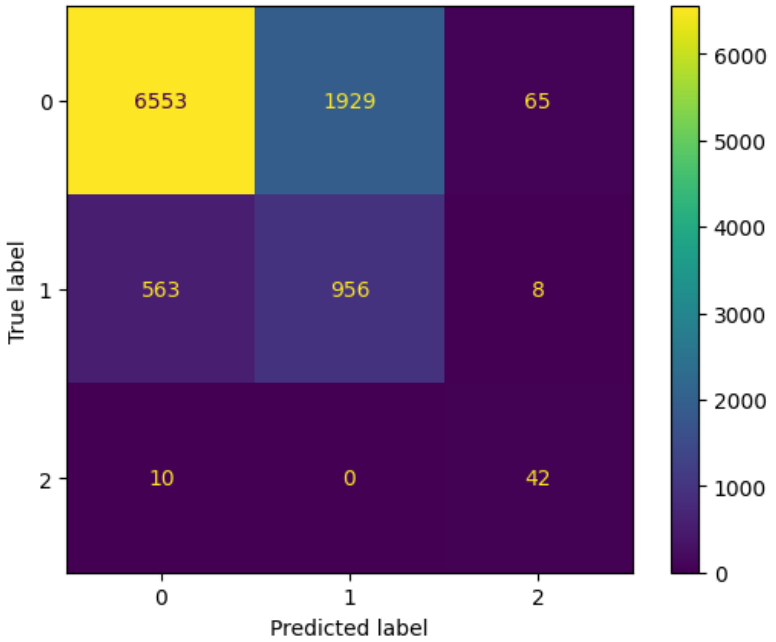
The model: XGBoost

- We decided to go with XGBoost, state-of-the-art model for what concerns Gradient Boosting techniques
- We did some trials to find the perfect model configuration:
  - XGBoost directly applied to our dataset
  - XGBoost with Sample Weights
  - XGBoost with Sample Weights and ADASYN resampling (applied GridSearch to find best hyperparameter configuration)
- For each trial, computed Classification Report and AUC score to evaluate the model performances
- Printing Feature Importance and Permutation Importance of features for each model

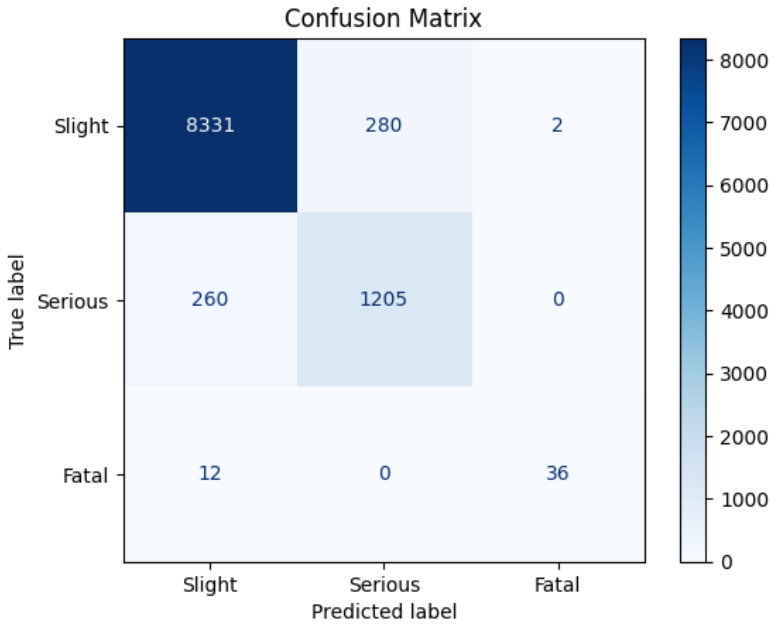
The models – Confusion Matrices



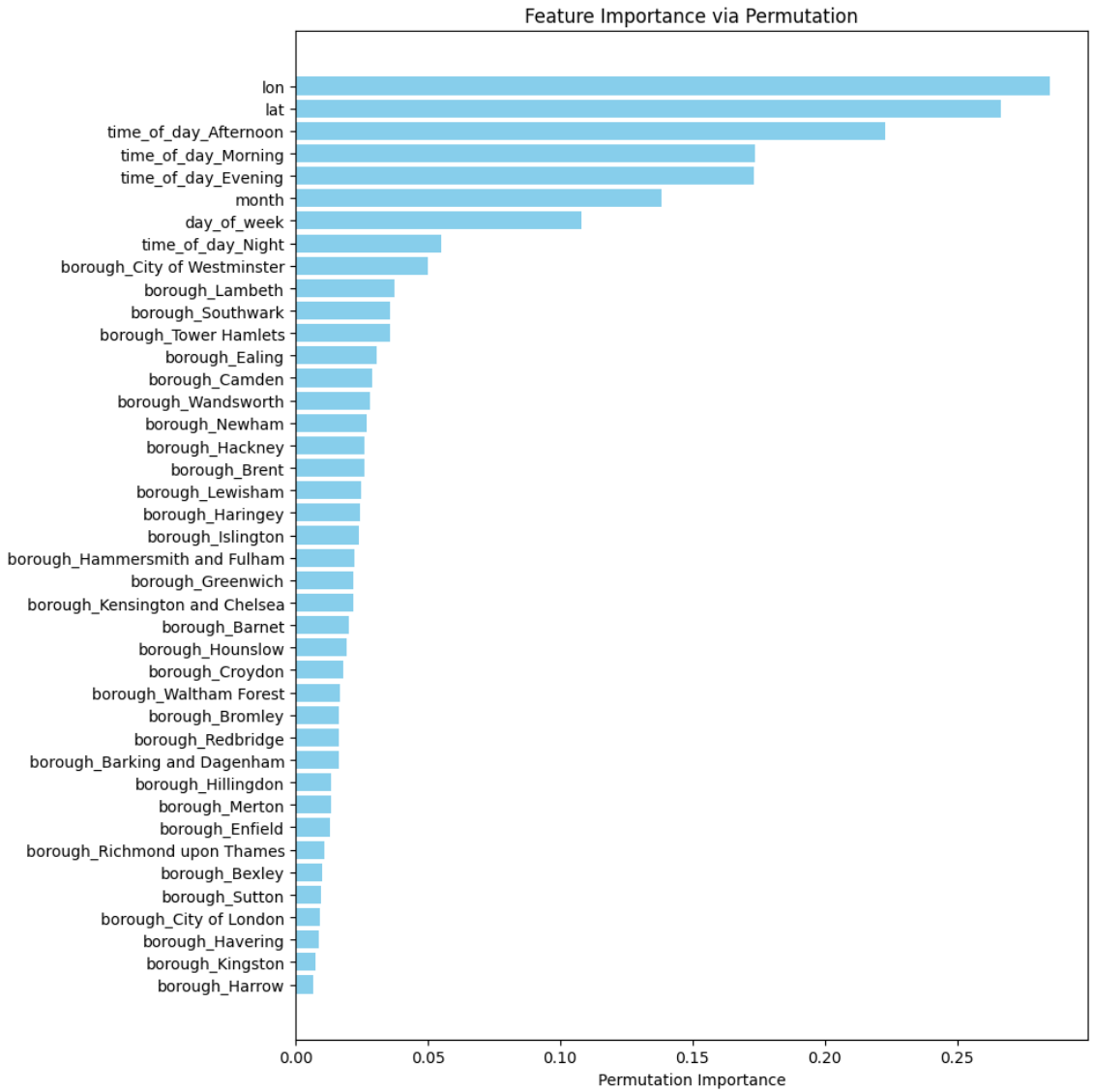
Default XGBoost
AUC score = 0.79

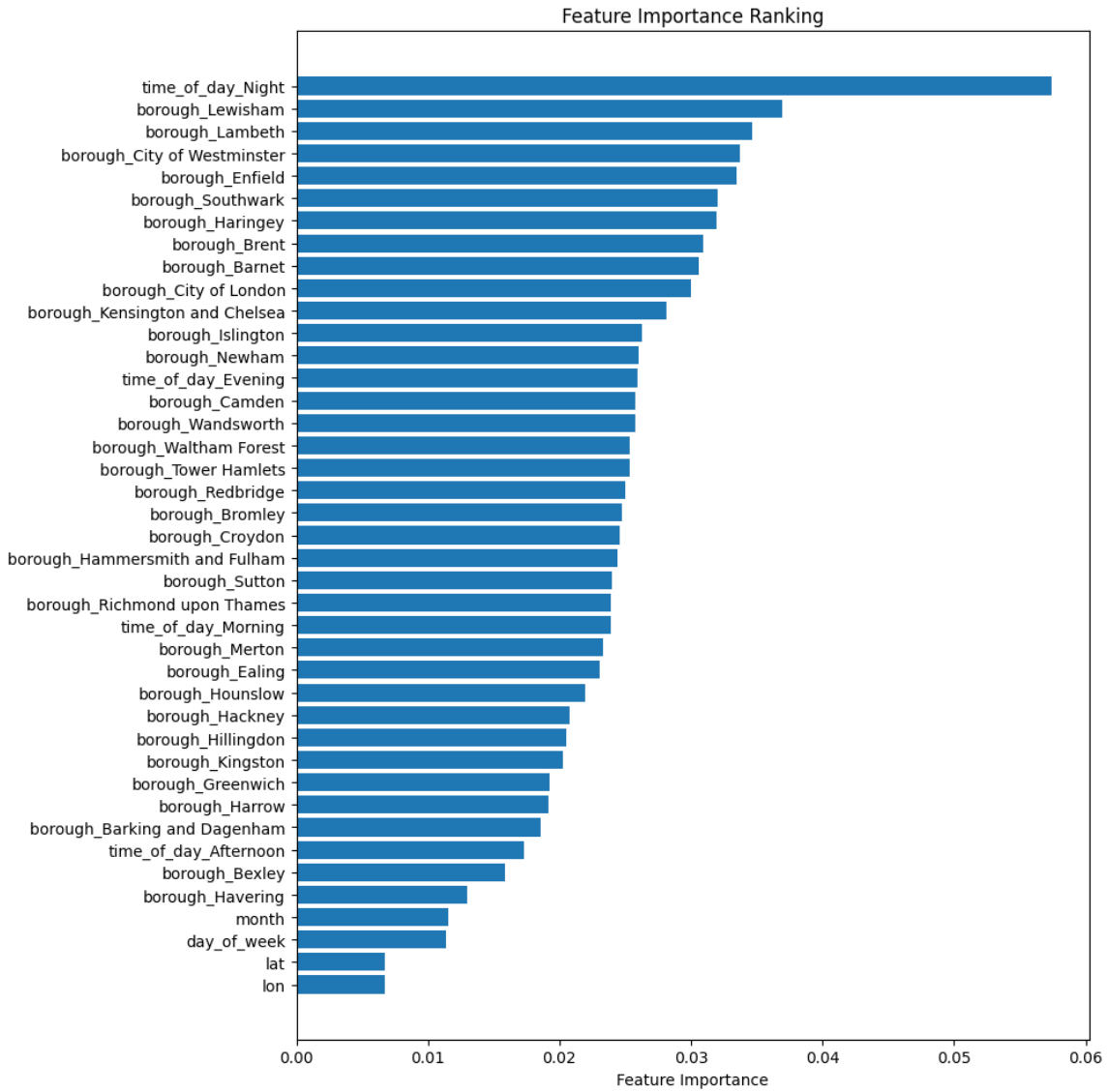


XGBoost w/ Sample Weights
AUC score = 0.80



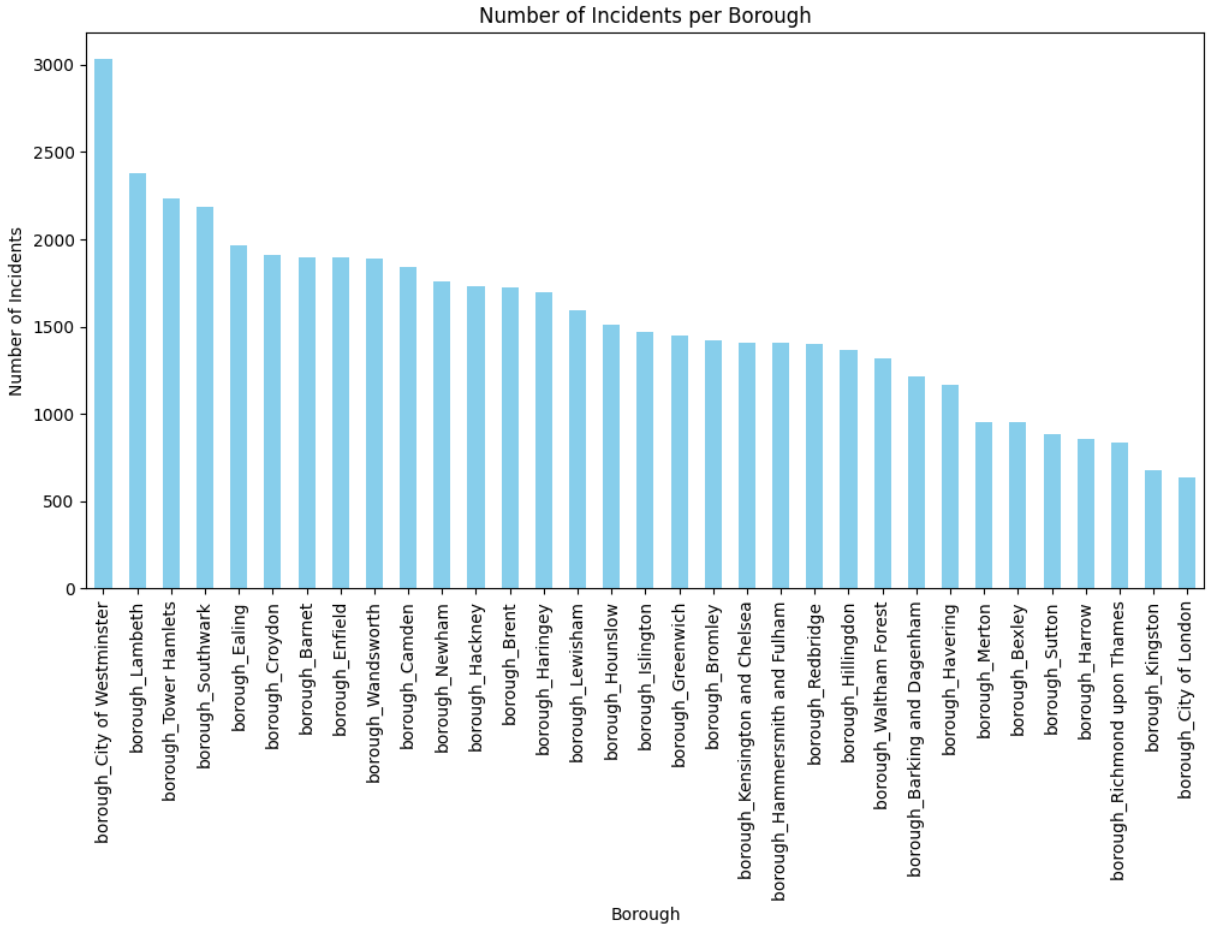
XGBoost w/ Sample Weights and ADASYN (GridSearch)
AUC score =0.92

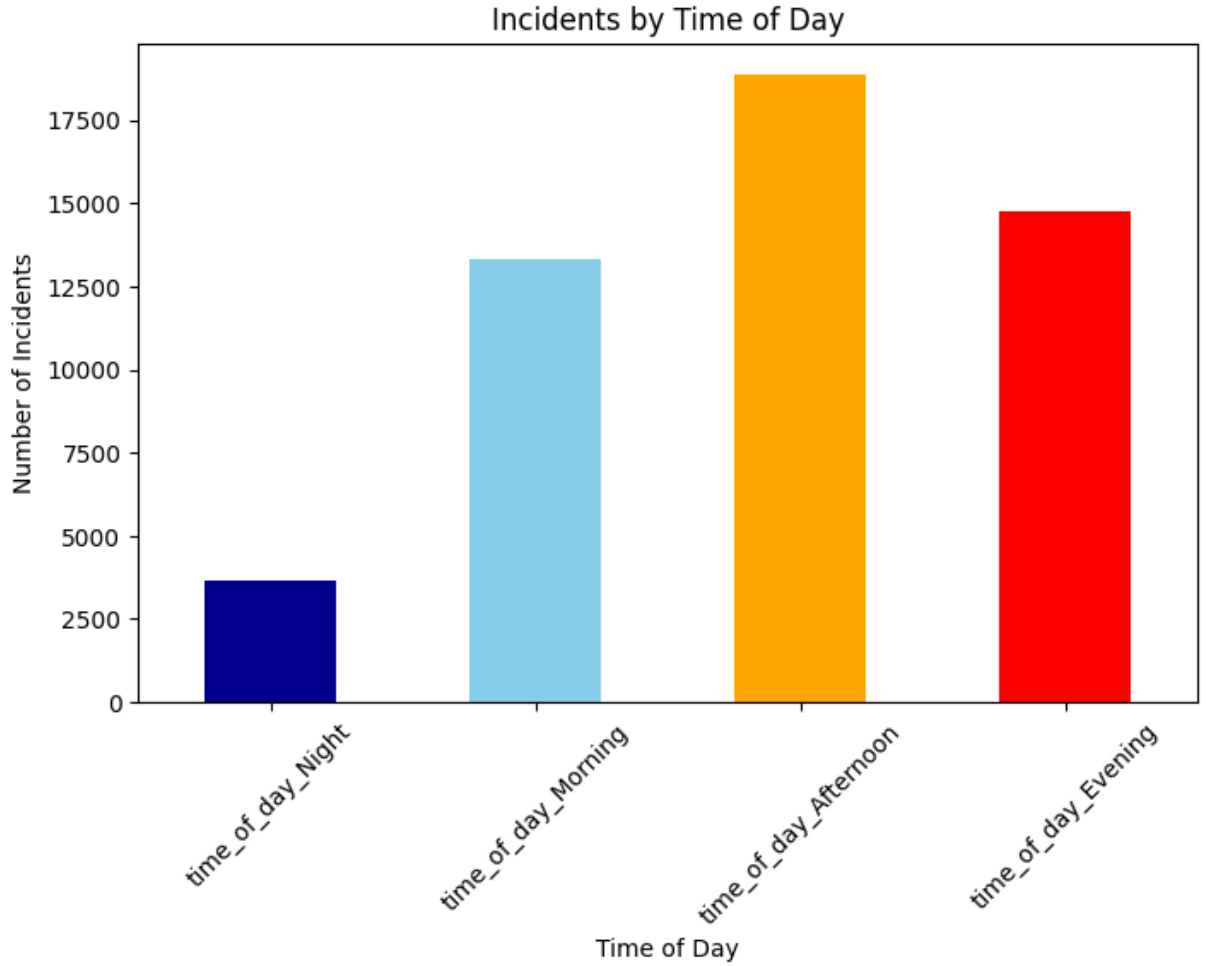




Permutation Importance and Feature Importance for the best model
Hyperparameters: {'learning_rate': 0.3, 'max_depth': 25, 'subsample': 0.8}

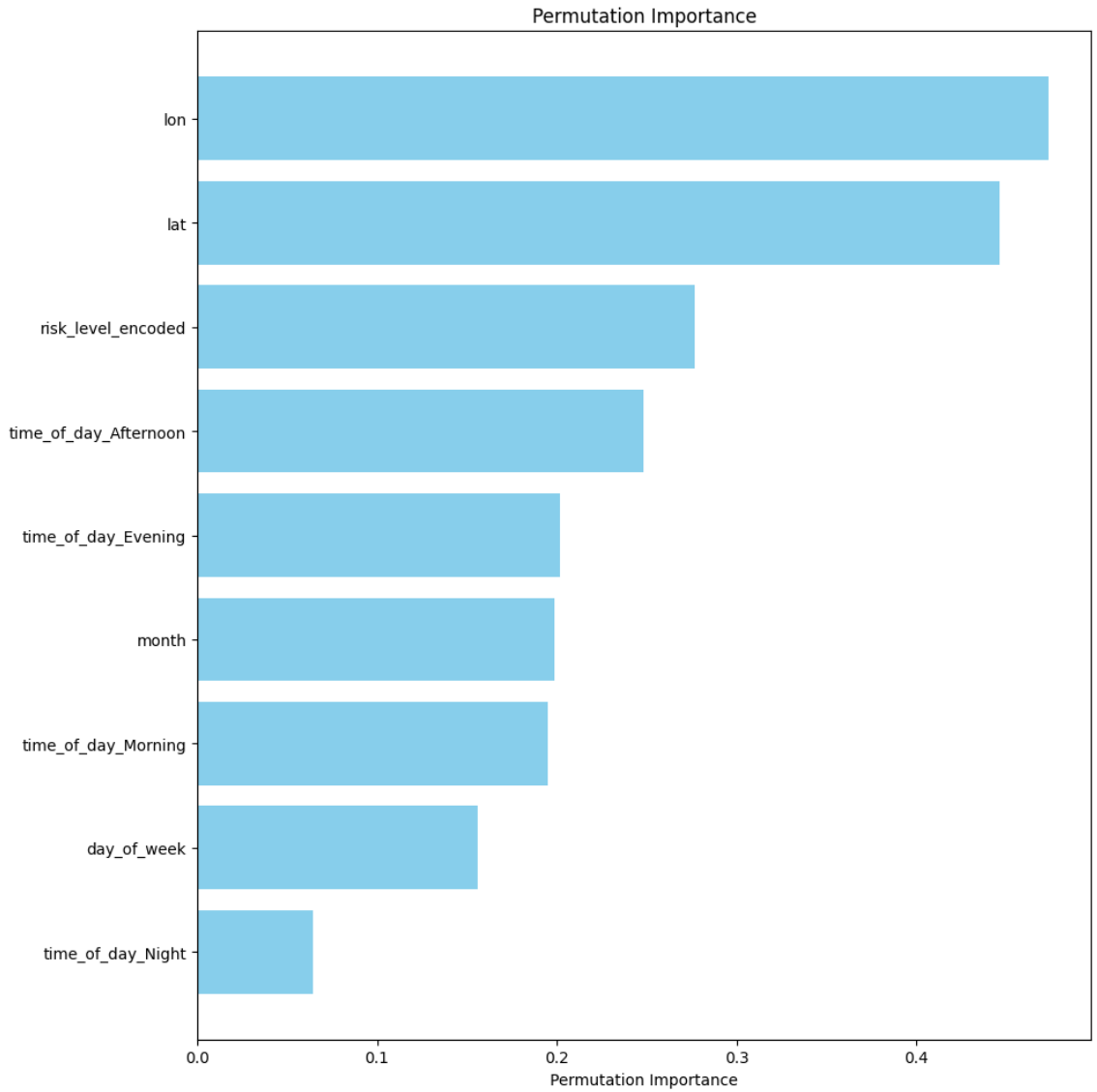
Our model reflects reality

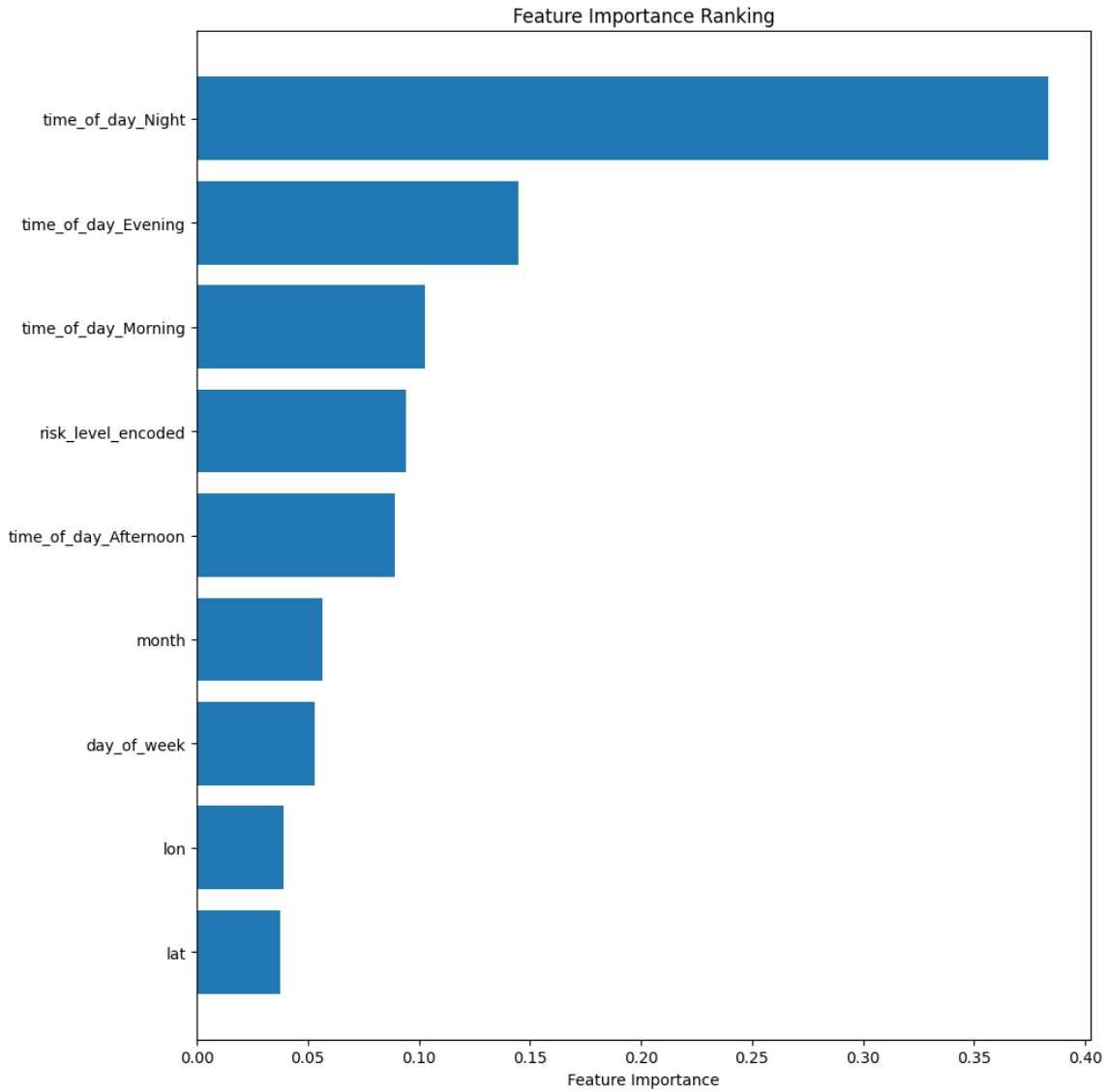




Risk Level

- Looking at the Permutation Importance graph, we can see that Borough variables have a very low relevance
- Splitting Borough in so many variables wasn't a good choice.
- Trying a different approach: redefining the boroughs as risk areas
  - Every borough will be assigned a risk level, from 0 (low) to 2 (high)
  - Risk_score = 0.3 * borough_tot_accidents + 0.7 * borough_avg_severity
- Defining the new dataframe and rerunning the classification model

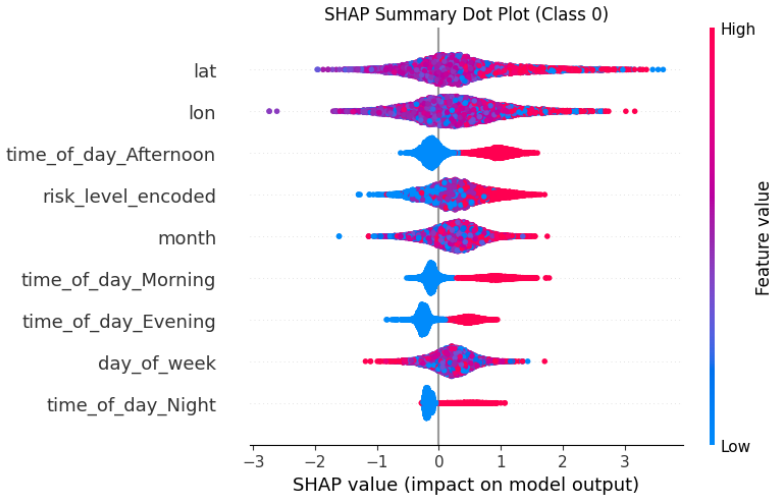

Permutation Importance
lon
lat
risk_level_encoded
time_of_day_Afternoon
time_of_day_Evening
month
time_of_day_Morning
day_of_week
time_of_day_Night
0.0
0.1
0.2
0.3
0.4
Permutation Importance

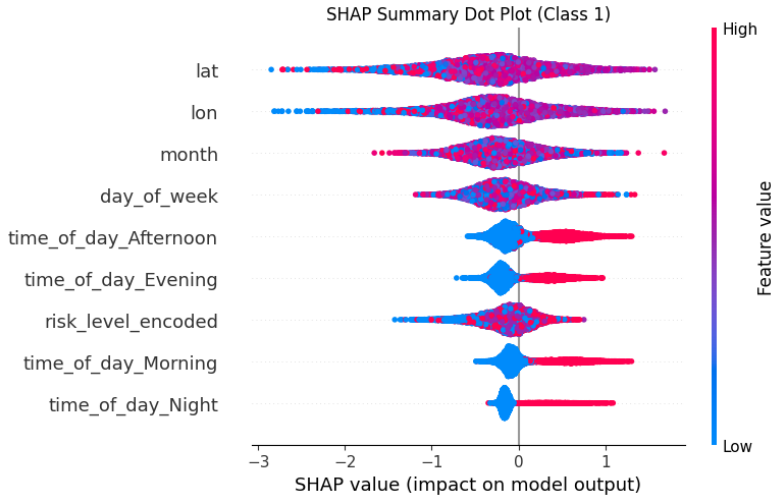

Feature Importance Ranking
time_of_day_Night
time_of_day_Evening
time_of_day_Morning
risk_level_encoded
time_of_day_Afternoon
month
day_of_week
lon
lat
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Feature Importance

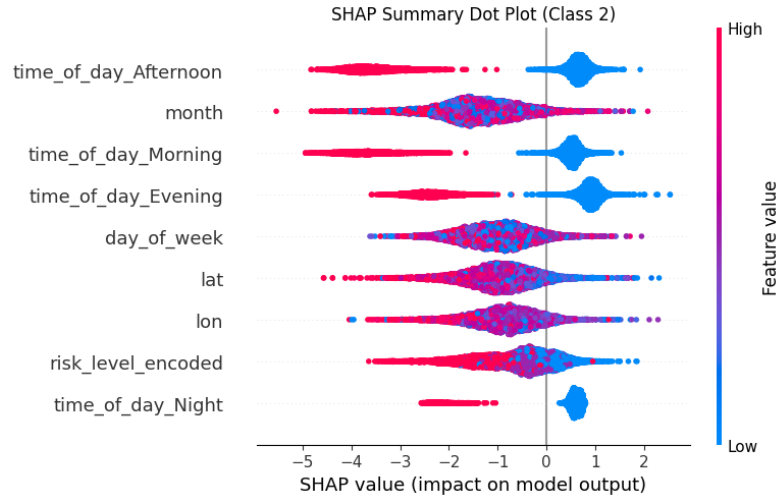
SHAP analysis

- To have more detailed analysis on how each feature influences the target variable, we used **SHAP**
- To visualize results we used a **SHAP dot plot**
- This allowed us to have a clearer view on each category and interpret our data

SHAP Analysis







Conclusion & Future Enhancements

- Location remains to be the most important feature, spread across the three categories.
- Time_of_day_afternoon also has a great importance, given it's one of the most trafficked time periods
- Model understands when and where of accidents, especially when identifying non-Fatal or light cases
- Future enhancements:
  - Improve on the behavior of risk_level_encoded: the number of accidents it's still too relevant compared to the severity
  - Drawing a heatmap on London based on the feature could be great for visualizing results

Project Work

Mohammed Oubia, Aaron Stephan Salazar Jaramillo

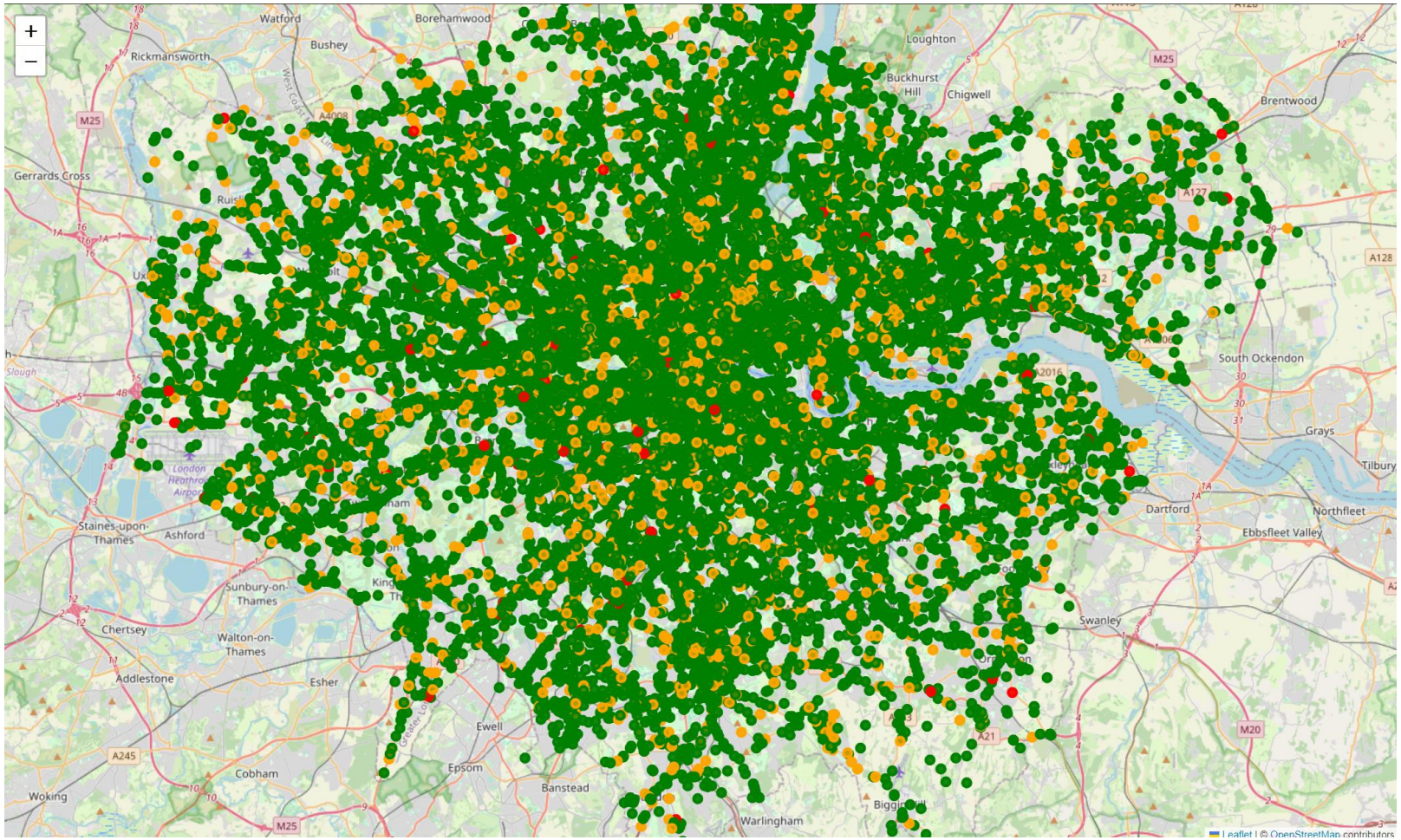
Extra-project work

- Accident Location Visualization with a Map.
- **LASSO (L1 Regularization)** used to select most relevant and Reduced dimensionality

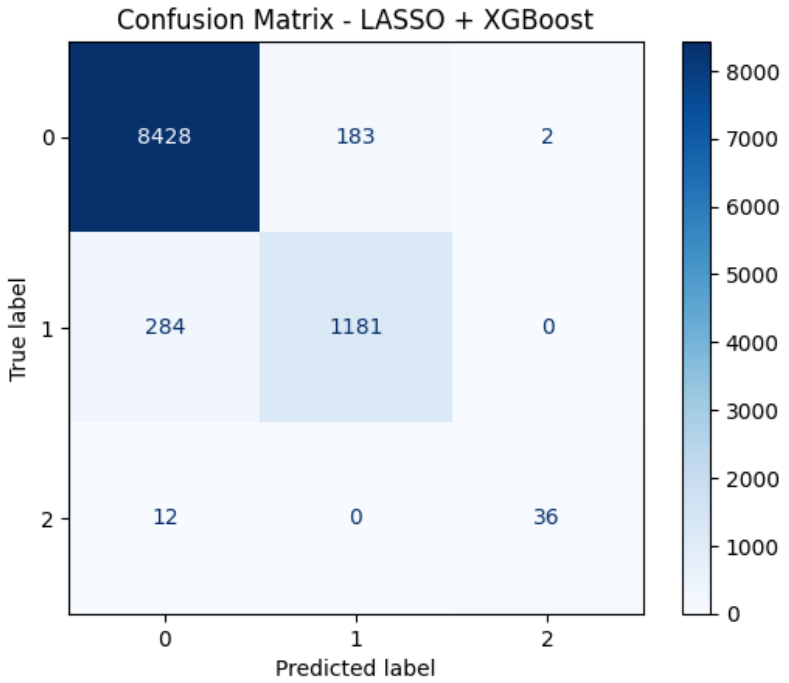
Removed redundant/noisy features.

- **XGBoost classifier** trained on LASSO-selected features with class-balanced ADASYN resampling.
- **SHAP** (SHapley Additive exPlanations).

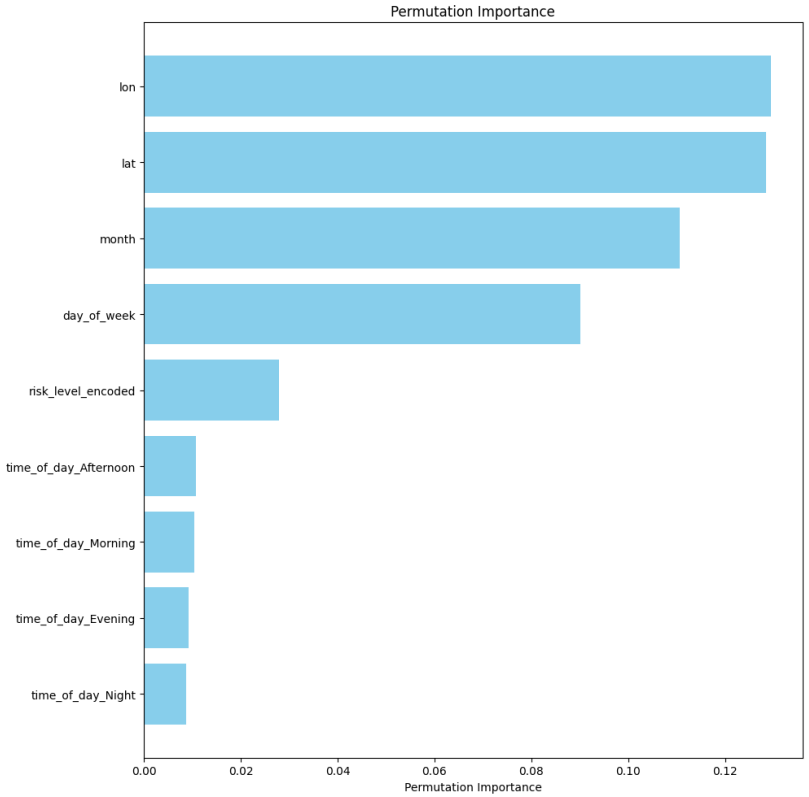
Map Visualization

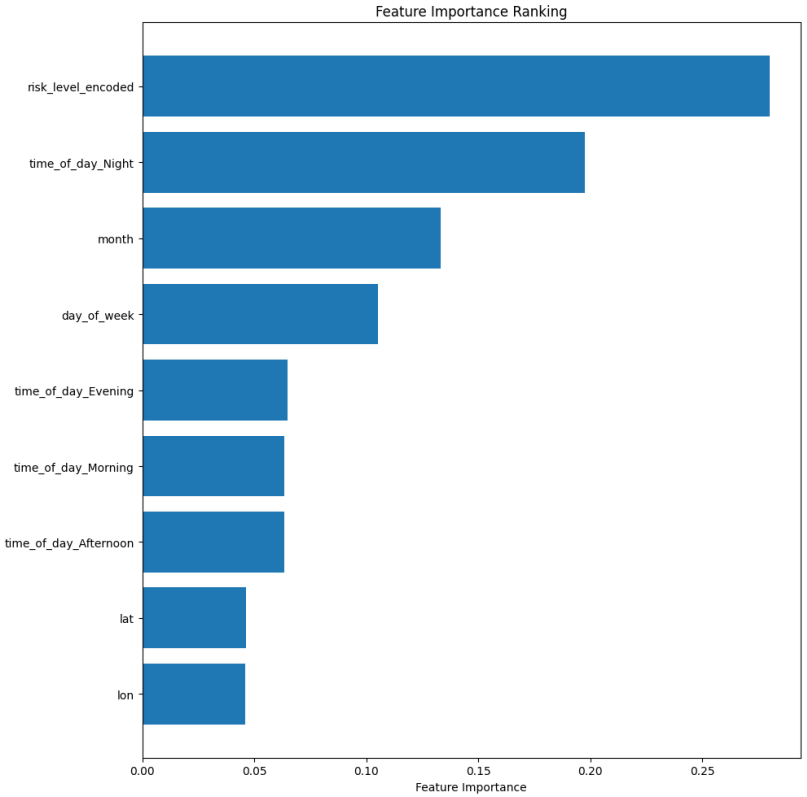


LASSO+XGBOOST Analysis

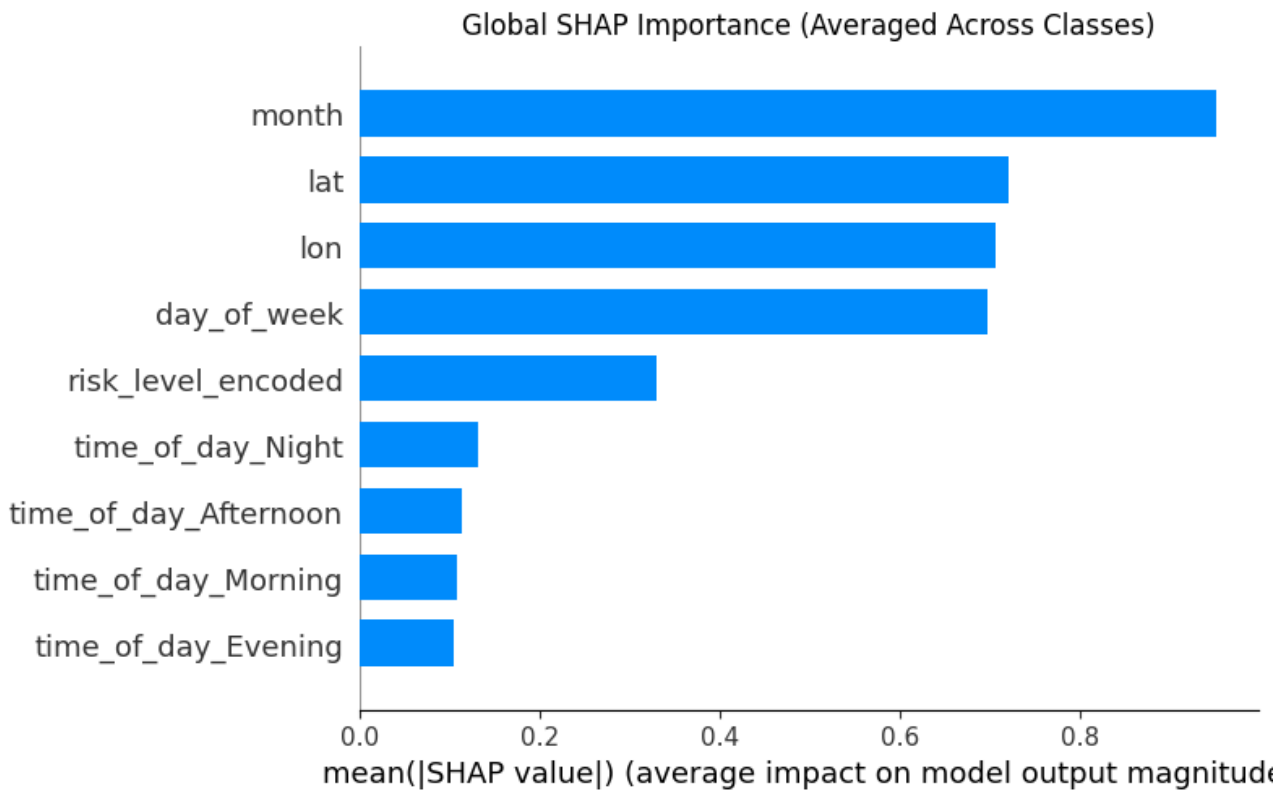


LASSO+ XGBoost
AUC score = 0.95





SHAP Analysis



Thanks for the attention