

|  |
| --- |
| **Predicting Road-Accident Risk with Machine Learning: Weather, Driver–Vehicle Factors, and Crash Hotspots** |
| DELVIN VALLOORAN JOSE  20039903 |
| Applied Research Project submitted in partial fulfilment of the requirements for the degree of MSc. Data Analytics  at Dublin Business School |
| Supervisor: **Dr Yalemisew Abgaz** |
| 26/08/2026 |

**DECLARATION**

‘I declare that this Applied Research Project that I have submitted to Dublin Business School for the award of [name your programme here] is the result of my own investigations, except where otherwise stated, where it is clearly acknowledged by references. Furthermore, this work has not been submitted for any other degree.’

Signed:Delvin Vallooran Jose

Student Number: 20039903

Date: 26/08/2025

**ACKNOWLEDGEMENT**

I would like to thank my supervisor for guiding me through the execution of this research. Without the guidance and feedback, I may not be able to complete this research. I would also like to thank all my professors for clearing the concepts of Machine Learning, Business Analytics and Statistical analysis.

**ABSTRACT**

Road accidents cluster at persistent black spots and are modulated by weather and driver/vehicle factors. This study develops a reproducible pipeline that combines variable-density hotspot detection (HDBSCAN as a exploratory and adopts 100m proximity rule used) with hour-level weather and selected driver/vehicle attributes from official U.K. STATS19. A unified dataset is constructed and 28 numeric features (coded categoricals) are engineered. Decision Tree, Random Forest, XGBoost, LightGBM and CatBoost classifiers are trained under stratified validation with leakage controls and probability calibration. Relative to structure-only baselines, adding hotspot and weather context improves discrimination and reliability; analysis highlights proximity to hotspots, precipitation, wind and light conditions as dominant signals, with driver/vehicle factors providing secondary lift. The output is a minimal Django interface that accepts location, time and driver/vehicle inputs, fetches hourly weather, and returns a probability and risk band for scenario queries and time-aware planning.

***Keywords: Road accidents, Blackspot detection, Weather data, Machine Learning, Accident prediction, HDBSCAN***

**Table of Contents**

[CHAPTER 1: INTRODUCTION 15](#_Toc207106678)

[Introduction 15](#_Toc207106679)

[Overview 16](#_Toc207106680)

[1.1 Background & Motivation 16](#_Toc207106681)

[1.2 Problem Statement 17](#_Toc207106682)

[1.3 Research Aim 17](#_Toc207106683)

[1.4 Objectives 17](#_Toc207106684)

[1.5 Research Questions 18](#_Toc207106685)

[1.6 Contributions 18](#_Toc207106686)

[1.7 Research Significance 19](#_Toc207106687)

[1.8 Structure of the Dissertation 19](#_Toc207106688)

[1.9 Summary 20](#_Toc207106689)

[CHAPTER 2: LITERATURE REVIEW 21](#_Toc207106690)

[Overview 21](#_Toc207106691)

[2.1 Introduction 21](#_Toc207106692)

[2.2 Black-spot Detection 22](#_Toc207106693)

[2.3 Weather & Crash Risk 23](#_Toc207106694)

[2.4 Driver/Vehicle Effects 23](#_Toc207106695)

[2.5 Predictive Modelling for Mixed Tabular Data 24](#_Toc207106696)

[2.6 Summary 25](#_Toc207106697)

[CHAPTER 3: METHODOLOGY 26](#_Toc207106698)

[Overview 26](#_Toc207106699)

[3.1 Research philosophy & design (abductive, design-science) 26](#_Toc207106700)

[3.2 Methodological architecture & strategy 26](#_Toc207106701)

[3.3 Data sources 28](#_Toc207106702)

[3.4 Research ethics & data governance 28](#_Toc207106703)

[3.5 Reproducibility & transparency 29](#_Toc207106704)

[3.6 Data preparation & integration 29](#_Toc207106705)

[3.7 Black-spot detection: exploration HDBSCAN, operational 100m grouping 30](#_Toc207106706)

[3.8 Weather & temporal feature engineering 32](#_Toc207106707)

[3.9 Modelling & Evaluation 33](#_Toc207106708)

[3.10 Machine Learning Algorithms 34](#_Toc207106709)

[3.11 Validation & error-analysis plan 35](#_Toc207106710)

[3.12 Summary 35](#_Toc207106711)

[CHAPTER 4: ARTEFACT 37](#_Toc207106712)

[Overview 37](#_Toc207106713)

[4.1 Data & feature schema(runtime) 37](#_Toc207106714)

[4.2 Model training & export 38](#_Toc207106715)

[4.3 Django architecture & loader 38](#_Toc207106716)

[4.4 User interface 38](#_Toc207106717)

[4.5 Deployment & run 40](#_Toc207106718)

[4.6 Runtime check & limitations 41](#_Toc207106719)

[4.7 Chapter summary 41](#_Toc207106720)

[CHAPTER 5: RESULTS AND FINDINGS 42](#_Toc207106721)

[Overview 42](#_Toc207106722)

[5.1 Exploratory patterns 42](#_Toc207106723)

[5.2 Model Performance 44](#_Toc207106724)

[5.3 Validation & generalisation checks 50](#_Toc207106725)

[5.4 Explainability 51](#_Toc207106726)

[5.5 Spatial context 52](#_Toc207106727)

[5.5 Chapter summary 53](#_Toc207106728)

[CHAPTER 6: CONCLUSIONS AND FUTURE WORKS 54](#_Toc207106729)

[Overview 54](#_Toc207106730)

[6.1 Conclusions 54](#_Toc207106731)

[6.2 Strengths and Weaknesses 55](#_Toc207106732)

[6.3 Theoretical contribution 55](#_Toc207106733)

[6.4 Practical Contribution 56](#_Toc207106734)

[6.3 Future Scope 56](#_Toc207106735)

[6.3 Chapter summary 57](#_Toc207106736)

[CHAPTER 7: DISCUSSION 58](#_Toc207106737)

[Overview 58](#_Toc207106738)

[7.1 Interpretation against prior literature 58](#_Toc207106739)

[7.2 Implications for practice and policy 58](#_Toc207106740)

[7.2 Limitations, generalisation and robustness 59](#_Toc207106741)

[7.3 Chapter summary 59](#_Toc207106742)

[CHAPTER 7: APPENDICES 60](#_Toc207106743)

[Appendix A — Data dictionary 60](#_Toc207106744)

[Appendix B — Additional figures and diagnostics 64](#_Toc207106745)

[Appendix C — User interface (UI) screenshots 66](#_Toc207106746)

[Appendix D — Model settings(deployed estimator) 68](#_Toc207106747)

[Appendix D.1 — Feature order 68](#_Toc207106748)

[Appendix D.2 — Preprocessing rules 68](#_Toc207106749)

[Appendix D.3 — Deployed model (XGBoost) 69](#_Toc207106750)

[Appendix D.4 — Training & valifation setup 70](#_Toc207106751)

[Appendix D.5 — Software environment(tested) 71](#_Toc207106752)

[Appendix D.6 — Serving contract 71](#_Toc207106753)

[Appendix E — Ethics and governance 71](#_Toc207106754)

[Appendix E.1 — Data sources & licensing 71](#_Toc207106755)

[Appendix E.2 — Personal data & privacy 72](#_Toc207106756)

[Appendix E.3 — Ethical risks & mitigations 72](#_Toc207106757)

[Appendix E.4 — Governance & security(when hosted) 73](#_Toc207106758)

[Appendix E.5 — DPIA note 74](#_Toc207106759)

[Appendix F — Reproducibility 74](#_Toc207106760)

[Appendix F.1 — Reproducibility 74](#_Toc207106761)

[Appendix F.2 — Environment 74](#_Toc207106762)

[Appendix F.3 — Recreate features & dataset 74](#_Toc207106763)

[Appendix F.4 — Train models & export the deployed estimator 75](#_Toc207106764)

[Appendix F.5 — Regenerate figures(Chapter 5 Appendix B) 75](#_Toc207106765)

[Appendix F.6 — Run the UI(Django) 76](#_Toc207106766)

[Appendix F.7 — Determinism & seeds 76](#_Toc207106767)

[Appendix F.8 — Hardware used (informative) 76](#_Toc207106768)

[Appendix F.9 — Hardware used (informative) 76](#_Toc207106769)

[REFERENCES 77](#_Toc207106770)

**List of Figures**

[Figure 3.1](#_Toc197618766) 27

[Pipeline overview: Data → Features → Models → UI.](#_Toc197618767) 27

[Figure 3.2](#_Toc197618768) 31

[Spatial clustering:(a) HDBSCAN (exploratory) (b)100 m proximity grouping. 17](#_Toc197618769)

[Figure 4.1](#_Toc197618770) 40

[Django scenario-scoring interface(blank form).](#_Toc197618771) 40

[Figure 4.2](#_Toc197618772) 40

[Django interface with completed inputs and prediction banner (class & probability).](#_Toc197618773) 40

[Figure 5.1](#_Toc197618774) 43

[Top-20 feature correlation heatmap(EDA)](#_Toc197618775) 43

[Figure 5.2](#_Toc197618776) 43

[Explained variance by principal component (PCA) 44](#_Toc197618777)

[Figure 5.3 44](#_Toc197618778)

**[PCA scatter (PC1-PC2) coloured by class](#_Toc197618779)**

**[44](#_Toc197618779)**

[Figure 5.4 46](#_Toc197618780)

[ROC curves for five classifiers 46](#_Toc197618781)

[Figure 5.5 46](#_Toc197618782)

[ROC curves for five classifiers 46](#_Toc197618783)

[Figure 5.6 47](#_Toc197618784)

[AUC by model (bar chart) 47](#_Toc197618785)

[Figure 5.7 47](#_Toc197618786)

[Accuracy by model 47](#_Toc197618787)

[Figure 5.8 48](#_Toc197618788)

[Precision by model 48](#_Toc197618789)

[Figure 5.9 48](#_Toc197618790)

[Recall by model 48](#_Toc197618791)

[Figure 5.10 49](#_Toc197618792)

[F1 score by model 49](#_Toc197618793)

[Figure 5.11 49](#_Toc197618794)

[Confusion matrix (XGBoost) 49](#_Toc197618795)

[Figure 5.12 40](#_Toc197618796)

[Feature importance(XGBoost, 28-feature schema). 52](#_Toc197618797)

[Figure B.1 64](#_Toc197618798)

[Confusion matrix Decision Tree(test set). 64](#_Toc197618799)

[Figure B.2 64](#_Toc197618800)

[Confusion matrix Random Forest (test set). 65](#_Toc197618801)

[Figure B.3 65](#_Toc197618802)

[Confusion matrix LightGBM(test set). 65](#_Toc197618803)

[Figure B.4 65](#_Toc197618804)

[Confusion matrix CatBoost(test set). 65](#_Toc197618805)

[Figure B.5 66](#_Toc197618806)

[Confusion matrix Over-fit. Tuned vs. OOF-test(test set). 66](#_Toc197618807)

[Figure C.1 66](#_Toc197618808)

[Accident Risk — Test: input form (blank state). 66](#_Toc197618809)

[Figure C.2 67](#_Toc197618810)

[Accident Risk — Test: input form (blank state). 67](#_Toc197618811)

[Figure C.3 68](#_Toc197618812)

[Accident Risk — Test: input form (blank state). 68](#_Toc197618813)

**List of Tables**

[Table 1: Structure of the Dissertation .](#_Toc197618832)19

[Table 2: Justification of Selecting ML algorithms](#_Toc197618833) 34

Table 3: Test-set metrics by model (AUC, Accuracy, Precision, Recall, F1)………………...50

Table 4: Generalisation checks comparing over-fitted and tuned runs, and OOF–test stability for the deployed configuration. Regularisation reduces gaps substantially; OOF≈Test indicates limited residual overfitting……………………………………………………………………51

Table A.1: Data dictionary for the deployed 28-features schema…………………………….60

Table D.1: Deployed XGBoost hyperparameters…………………………………………….69

# CHAPTER 1: INTRODUCTION

# Introduction

Road traffic collisions impose significant human and economic costs in the U.K. Empirical evidence shows that crashes concentrate at specific locations (“black spots”) and that risk is further influenced by time-varying weather and characteristics of drivers and vehicles. However, existing approaches are fragmented: hotspot mapping often omits meteorology and driver/vehicle context, whereas predictive models frequently ignore spatial autocorrelation and variable density between urban and rural networks. These gaps hinder the production of reliable, actionable risk signals.

This project addresses the problem by building a fused, reproducible pipeline that (i) detects and maintains black-spot clusters robust to varying densities, (ii) links official collision and vehicle tables to hour-level historical weather, (iii) engineers temporal, spatial and meteorological features, and (iv) trains and compares tree-based classifiers with calibration and explanations. The work answers three questions: which model best predicts accident risk with spatial–weather features; how weather patterns interact with black-spot proximity; and whether adding driver/vehicle attributes improves discrimination and reliability.

The study contributes: a hotspot-aware, weather-conditioned modelling framework; a comparative evaluation of five ensemble and baseline classifiers; and a lightweight Django artefact that accepts location, time and coded driver/vehicle inputs, fetches hourly weather, and returns a calibrated probability with a risk band for scenario queries. The remainder of the report covers related work (Chapter 2), methodology and research design (Chapter 3), implementation and artefact (Chapter 4), findings and discussion (Chapter 5), and conclusions with future work (Chapter 6).

## Overview

This chapter frames the study and sets the direction. It outlines the problem context, states the aim, and defines objectives and research questions. It summarises the contributions and practical significance, explains why a fused spatial–weather–driver/vehicle approach is needed, and signposts the dissertation structure that follows.

## 1.1 Background & Motivation

**Background.** Road traffic collisions impose substantial human and economic costs. They also exhibit clear spatial patterns: incidents concentrate at persistent “black spots” shaped by road geometry, sight lines, traffic mix, and behavioural factors (Perrels et al., 2015; Lobo et al., 2019). Adverse weather—rain, fog, snow, —reduces surface friction and driver perception, with night-time visibility a recurring issue. While hotspot analysis and weather-only models each explain part of the risk, treating them in isolation misses important interactions. This motivates an integrated, predictive approach that combines location, weather, vehicle, and human factors to support proactive mitigation and timely alerts (Wang et al., 2023; Szenasi et al., 2022.

**Motivation.** A formative case for this project was a road section in a local town that experienced crashes on the order of twice per month, predominantly at night. Constrained sight distance, poor lighting and through-traffic unfamiliar with the area appeared to contribute, alongside the local road layout. The case underlines the need for evidence-driven tools that can pinpoint such sites, quantify how visibility and design interact with weather, and generate timely, interpretable risk signals to support targeted countermeasures and traveller information.

## 1.2 Problem Statement

Current U.K. accident-risk approaches are fragmented. Spatial hotspot maps seldom incorporate hour-level meteorology and driver/vehicle attributes; conversely, predictive models frequently ignore spatial autocorrelation and variable density between urban and rural roads. This limits discrimination, calibration and interpretability, and makes operational use difficult. The need is for decision-ready, calibrated probabilities that integrate spatial priors (black spots) with time-varying weather and driver/vehicle context (Zhao, Lin and He, 2022). The aim is to design and evaluate a reproducible pipeline that links STATS19 to hourly weather, engineers features, and compares tree-based classifiers, quantifying the added value of hotspot and weather features over structure-only baselines, and delivering a minimal interface for scenario queries at a given location–time–vehicle profile.

## Research Aim

To design and evaluate a reproducible, hotspot-aware, weather-conditioned modelling pipeline that produces calibrated accident-risk probabilities and a minimal interface for scenario use in the U.K.

## 1.4 Objectives

The study first constructs a unified dataset by linking STATS19 collisions and vehicles to hour-level Open-Meteo weather, engineering twenty-eight coded numeric features and detecting black spots with HDBSCAN alongside a 100 m baseline. It then trains and compares Decision Tree, Random Forest, XGBoost, LightGBM and CatBoost under stratified validation, reporting AUC, F1, precision/recall and calibration, and quantifying the added value of hotspot and weather features against structure-only baselines via ablation analysis. Finally, a minimal Django artefact is delivered that fetches hourly weather, accepts location–time and driver/vehicle inputs, loads the trained model on the twenty-eight-feature schema and returns a probability and risk band; generalisation checks (out-of-fold versus test) and operating thresholds are documented (Breiman, L. (20016); Chen, T. and Guestrin, C. (2016); Ke, G. et al. (2017); Dorogush, A.V., Ershov, V. and Gulin, A. (2018)).

## 1.5 Research Questions

RQ1: Which model best predicts accident risk given spatial and weather features?

RQ2: How do specific weather patterns interact with black‑spot proximity to affect risk?

RQ3: Does adding driver/vehicle history measurably improve discrimination and calibration?

## 1.6 Contributions

The report contributes along three fronts. First, it develops a fused, leakage-aware pipeline that uses exploratory hotspot analysis (HDBSCAN) to understand spatial concentration and adopts a 100 m proximity rule operationally for grouping/sampling, while engineering a compact 28-feature schema for modelling, using coded categoricals consistent between training and deployment. The pipeline integrates exploratory hotspot analysis with hour-level weather and coded driver/vehicle attributes; a 100 m proximity rule is used operationally for grouping and sampling, while the deployed 28-feature model focuses on weather–temporal–driver/vehicle signals. Second, it provides an empirical comparison of Decision Tree, Random Forest, XGBoost, LightGBM and CatBoost on the unified dataset, reporting accuracy, precision/recall, F1 and AUC, and analysing feature importance to explain dominant signals. Third, it delivers a minimal Django artefact that fetches hourly weather, accepts location–time and driver/vehicle inputs, loads a non-Pipeline estimator trained on the 28 columns, and returns a calibrated probability on the 28-feature schema; hotspot flags are not used as inputs in the deployed model.

## 1.7 Research Significance

This This study is significant in three respects. **Methodologically**, it unifies variable-density hotspot detection with hour-level meteorology and coded driver/vehicle attributes in a single, leakage-aware pipeline, producing calibrated probabilities rather than uncalibrated scores, hotspot analysis supports curation and sampling; the deployed classifier focuses on weather–temporal–driver/vehicle signals. The approach addresses fragmentation in prior work by combining spatial priors with time-varying context and validating generalisation through out-of-fold and test-set checks. **Practically**, the output is an operational, minimal Django interface that accepts location–time and driver/vehicle inputs, fetches hourly weather, and returns a probability with a descriptive risk band, allowing local authorities to prioritise micro-hotspots and schedule time-aware interventions. **Reproducibility & transferability** are supported by a compact 28-feature schema with numeric codes, consistent units, and a non-Pipeline model loader, enabling straightforward redeployment across regions using official STATS19 and public weather archives while maintaining privacy-aware presentation.

## 1.8 Structure of the Dissertation

##### Table 1: Structure of the Dissertation

|  |  |
| --- | --- |
| ***Chapters*** | ***Significance*** |
| 1.Introduction | Chapter 1 introduces the problem, aim and objectives, research questions, contributions, significance and structure. |
| 2.Literature Review | Chapter 2 reviews prior work on black-spot detection, weather–risk modelling and tabular classifiers; it synthesises gaps and positions the study. |
| 3.Methodology | Chapter 3 details the methodology: research philosophy and abductive design; data sources; ethics and governance; reproducibility; data preparation and integration; hotspot detection; weather and feature engineering; modelling and evaluation; validation and error analysis. |
| 4.Artefact | Chapter 4 describes the artefact: dataset schema, model training and export, Django architecture, user interface and deployment notes. |
| 5.Result Analysis | Chapter 5 reports the result analysis: exploratory patterns, comparative metrics for five models (AUC, F1, precision/recall), ROC–PR curves, calibration, confusion matrices and feature importance. |
| 6.Findings and Discussion | Chapter 6 discusses the findings in relation to the literature and outlines practical implications for planning and operations. |
| 7.Conclusion and Recommendations | Chapter 7 concludes and recommends future work. Appendices provide the data dictionary, hyperparameters and split settings, UI screenshots and additional figures. |

## 1.9 Summary

Chapter 1 summarises the problem, aim and objectives, and why a joined-up approach is needed. It frames collision risk as spatially clustered and weather-sensitive, and motivates integrating location, hour-level meteorology, and driver/vehicle attributes. It states three research questions on prediction, added value of hotspots and weather, and operational usability. It also outlines the project’s contributions—a reproducible pipeline, comparative modelling with calibrated outputs, and a lightweight decision-support interface—and notes ethical safeguards and reproducibility. The chapter closes by signposting the remaining structure of the dissertation.

# CHAPTER 2: LITERATURE REVIEW

## Overview

Chapter 2 surveys prior work on crash hot-spots, weather–risk relationships, and tabular machine-learning for safety prediction. The goal is to identify gaps that motivate a fused, hotspot-aware and weather-conditioned pipeline. First, approaches to spatial concentration are reviewed, from kernel density estimation and sliding windows to density-based clustering (Silverman, 1986). Strengths and limitations are assessed in the context of variable urban–rural densities and irregular cluster shapes. Second, literature on meteorology and crash likelihood is summarised, including the roles of precipitation, wind, visibility and darkness, and the need for hour-level alignment. Third, evidence on classification methods for mixed tabular data is examined, with attention to class imbalance, calibration and explainability. The review closes by articulating the unresolved issues: separation of mapping and prediction, limited treatment of spatial autocorrelation, and a lack of calibrated, operational outputs. These gaps position the present study’s integrated design and minimal, deployable artefact.

## 2.1 Introduction

This chapter reviews four areas central to the study: black-spot detection, weather–risk relationships, driver/vehicle factors, and predictive models for mixed tabular data. Searches (2015–2025) in Google Scholar and transport safety journals prioritised peer-reviewed sources and official guidance. A recurring gap emerges: hotspot maps rarely incorporate time-varying meteorology and driver/vehicle context; many predictive models ignore spatial dependence and variable urban–rural densities; and calibrated, operational outputs are uncommon. Sections 2.2–2.5 summarise these strands and motivate the integrated, hotspot-aware and weather-conditioned pipeline deployed in this project.

## 2.2 Black-spot Detection

Black-spot analysis traditionally relies on kernel density estimation (Silverman, 1986) or fixed windows/grids to highlight crash concentrations. These techniques are simple and produce smooth surfaces but can over-smooth risk, impose arbitrary boundaries, and struggle with irregular cluster shapes or sharp urban–rural density changes. Density-based clustering addresses several of these issues. DBSCAN can recover arbitrarily shaped clusters and treat isolated collisions as noise; (Alotaibi, 2018) however, a single global neighbourhood radius is difficult to tune when dense inner-city streets and sparse rural links coexist (Szenasi et al., 2022). HDBSCAN extends DBSCAN by building a hierarchy of clusters and selecting the most stable structures, (Wang et al., 2023) which improves robustness to variable density and reduces sensitivity to parameter choice. For operational road-safety work, this yields clusters that respect junction approaches and linear features without forcing grids. For this project, HDBSCAN was used in the exploratory phase to verify that collisions form variable-density clusters and to guide the choice of a practical grouping radius. The operational dataset then adopted a **100 m proximity rule** to aggregate micro-hotspots around junction approaches, (Iqbal et al., 2020) which reduced runtime and memory while preserving local concentration. Empirically, the 100 m groups aligned with the dense cores highlighted by HDBSCAN in urban corridors, with known trade-offs in very sparse areas. This pragmatic choice yielded a stable spatial prior for curation and sampling; limitations are acknowledged in Chapter 5.

## 2.3 Weather & Crash Risk

Weather conditions modulate crash likelihood through multiple pathways: precipitation reduces surface friction and lengthens stopping distances; (Perrels et al., 2015) fog and low cloud impair visibility; wind direction, speed and gusts destabilise vehicles and affect lane-keeping; temperature and dew-point interactions drive icing and mist; and cloud cover interacts with daylight to alter detection and gap-acceptance. Effects are highly time-dependent: the same location can alternate between low and high risk within hours as rain bands pass or darkness falls (Lobo et al., 2019). Measurement and alignment matter; coarse daily summaries obscure short-lived hazards and induce temporal mismatch. Consequently, hour-level meteorology tied to the exact scenario time is preferable (Becker, Rust and Ulbrich, 2022). In this study, the weather feature set comprises surface pressure, temperature and apparent temperature, dew-point and relative humidity, layered cloud cover, precipitation indicators (rain, snowfall), and wind direction/speed/gusts, joined at the scenario hour. Temporal context is encoded via time\_group (1–24) and day\_of\_week (1–7). This framing captures both instantaneous conditions and routine rhythms (rush hours, night-time) and provides signals that tree-based classifiers can exploit without strong linearity assumptions.

## 2.4 Driver/Vehicle Effects

Crash risk varies with characteristics of drivers and vehicles as well as with exposure. Age effects are typically non-linear: younger drivers show elevated risk linked to inexperience, (McCartt et al., 2009) while very old drivers are affected by declines in perception and reaction time. Sex-of-driver, journey purpose and distance banding capture behavioural context (e.g., commuting, business use, school travel) that shapes time–place exposure. Area deprivation (e.g., IMD decile) is a proxy for road environment and vehicle fleet differences rather than an intrinsic attribute of individuals, and should be interpreted accordingly. Vehicle factors influence both crash involvement and injury outcomes: vehicle type (car, goods, motorcycle) has distinct dynamics; engine capacity and power relate to acceleration profiles; vehicle age approximates the presence of safety technology; (Blows et al., 2003) and towing or articulation changes handling margins. Home-area type (urban/rural) interacts with network geometry and typical speeds. In this project, these variables are encoded as numeric codes with “unknown” explicitly represented, used as contextual predictors alongside weather and time. The aim is to obtain calibrated probabilities without profiling individuals, and to support planning decisions rather than enforcement.

## 2.5 Predictive Modelling for Mixed Tabular Data

Predictive modelling for mixed tabular data is well served by tree ensembles, which capture non-linearities and interactions. A single decision tree is interpretable but high variance; (Zhao, Lin and He, 2022) bagging (Random Forest) reduces variance, whereas gradient boosting fits many shallow trees to reduce bias. Implementations such as XGBoost, LightGBM and CatBoost are widely used. In this study, categories are encoded as integer codes (−1 for unknown) and the deployed estimator consumes a fixed 28-feature numeric schema, avoiding mismatch with one-hot pipelines.

Imbalance and evaluation require care. Stratified splits are combined with threshold selection. Discrimination is reported with ROC-AUC together with precision, recall and F1; reliability is assessed through probability calibration (isotonic or Platt). Validation follows an out-of-fold protocol plus a held-out test set; generalisation is judged by OOF–test gaps and by tightening regularisation (depth, learning rate, subsampling and L1/L2) where needed to curb overfitting.

Interpretability relies on model-native importance and targeted error analysis. The deployed model emphasises weather and temporal predictors with driver/vehicle context; hotspot analysis informs curation and sampling rather than serving as direct predictors.

## 2.6 Summary

Chapter 2 reviews four strands: black-spot detection methods, weather–risk evidence, driver/vehicle effects, and tree-based modelling for mixed tabular data. It highlights limitations in common practice: hotspot maps rarely include time-varying weather or vehicle/driver context, while predictive models often ignore spatial dependence and variable density. The review synthesises these gaps to justify an integrated pipeline that combines clustering, weather enrichment and interpretable ensembles. It concludes by restating the research questions the study addresses.

# CHAPTER 3: METHODOLOGY

## Overview

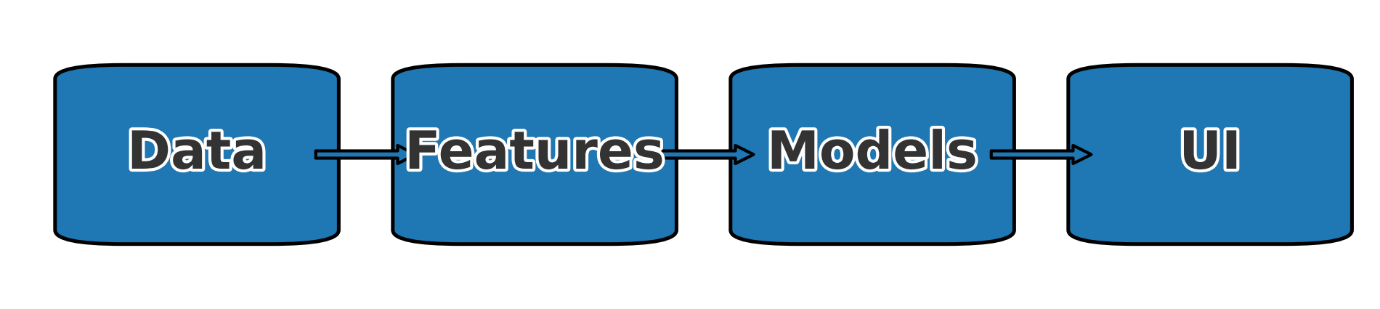
This chapter details the methodology used to build and evaluate the artefact. It sets out the research philosophy and design, data sources and ethics, reproducibility measures, and the end-to-end procedure from data preparation to feature engineering. It explains exploratory HDBSCAN versus the operational 100 m grouping, the modelling and evaluation plan, hyperparameter strategy, and validation and error-analysis steps.

## 3.1 Research philosophy & design (abductive, design-science)

The study adopts a pragmatist stance and an abductive, design-science approach. Observed patterns—spatial clustering of collisions and weather-driven variability—were used to form hypotheses, which were iteratively tested and refined to deliver a working artefact. The objective is predictive accuracy and calibrated probabilities suitable for operational use rather than causal estimation of specific interventions. The design emphasises a reproducible pipeline from public data to an interface that answers scenario queries at a given location, time and driver/vehicle profile.

## 3.2 Methodological architecture & strategy

Methodologically, the project follows a data-driven predictive strategy: curate official collision and vehicle tables, align them with hour-level meteorology, construct a fixed numeric feature schema, and evaluate tree-based classifiers under stratified validation. Exploratory hotspot analysis was performed to understand spatial concentration and to inform pragmatic grouping, while the deployed classifier focuses on weather, temporal and driver/vehicle context. The architecture separates concerns into data acquisition, feature engineering, modelling, validation and deployment, with clear interfaces so that components can be replaced without breaking the pipeline.



##### Figure 3.1: Pipeline overview: Data → Features → Models → UI.

##### 3.2.1: Materials, tools & technologies

The implementation uses Python 3.x with pandas and numpy for data handling, scikit-learn for validation utilities and baseline models, XGBoost and LightGBM for gradient-boosted trees, and Django with requests for the scenario-scoring interface and weather retrieval. Experiments ran on a standard CPU workstation; GPU acceleration was unnecessary because the histogram tree back-ends are efficient on tabular data. Random seeds are fixed for splits and model initialisation. Weather units are pinned to match training (km/h wind, hPa pressure, °C temperature, mm precipitation) to avoid train–serve skew.

##### 3.2.2: Procedure(end-to-end workflow)

The workflow begins with prototyping joins and features, followed by migration to the official STATS19 collision and vehicle tables. Exploratory clustering with HDBSCAN confirms variable density and informs the choice of an operational 100 m proximity rule to aggregate micro-hotspots and control runtime on the full dataset. Hour-level weather is then joined at each scenario location and time using the archive endpoint with pinned units. Non-accident examples are created by sampling date–time points at the same locations while excluding known crash timestamps; these are enriched with weather in the same way as the accident records. Vehicle and driver attributes are merged on the accident reference, invalid values are removed, and classes are balanced by random trimming. Categorical fields are encoded as integer codes with −1 for “Unknown,” yielding a fixed, 28-feature numeric schema. Models are trained under stratified validation, calibrated where necessary, and the final estimator is exported for use in the Django interface, which fetches weather and produces a probability and risk band. See **Figure 3.1** for an overview.

## 3.3 Data sources

Official U.K. STATS19 collision records were used together with the associated vehicle tables that contain driver and vehicle attributes (e.g., driver age, journey purpose, vehicle type and engine capacity). Hour-level weather was retrieved for scenario locations from Open-Meteo’s archive, with units pinned to training conventions (km/h wind speed, hPa pressure, °C temperature, mm precipitation). Early exploration used a Kaggle variant to prototype joins and features; the final analysis relies on the official releases and the weather archive. All sources are secondary and non-identifiable.

## 3.4 Research ethics & data governance

Only variables necessary for modelling are retained; no personally identifiable information is processed or displayed. Outputs are probabilities and summary indicators rather than person-level labels. The non-accident class is generated by sampling date–time points at accident locations while excluding known crash timestamps, which approximates exposure without tracking individuals. Risk communication avoids profiling subgroups and emphasises uncertainty and calibration. Validation and documentation are used to reduce over-interpretation, and any deployment should include threshold governance, periodic review and rollback criteria if drift or unintended consequences are detected.

## 3.5 Reproducibility & transparency

Reproducibility is supported by a fixed **28-feature** numeric schema with coded categoricals (−1 for unknown) and a recorded column order. Weather units are pinned to match training. Random seeds and stratified splits are recorded; performance is reported as out-of-fold estimates alongside a hold-out test set. The saved model is a single estimator (non-Pipeline) trained on exactly those 28 columns, avoiding feature-matrix mismatch at runtime. Input construction, imputation rules and threshold selection are documented, and the scripts that generate figures and tables read from the produced datasets to minimise manual steps.

##### 3.5.1: Software, hardware & versions

Experiments were executed in Python 3.x using pandas, numpy, scikit-learn, xgboost and lightgbm. Random seeds are fixed for splits and model initialisation. Library versions and key flags (e.g., tree\_method="hist", pinned weather units) are documented to enable like-for-like replication.

## 3.6 Data preparation & integration

Collision and vehicle tables are joined on the accident reference to add driver and vehicle attributes. Weather is then attached at the scenario hour for each location using the archive endpoint with pinned units. To create negatives, random date–time samples are generated at the same locations while excluding actual accident timestamps; these are enriched with weather in the same manner, producing non\_accident\_weather\_balanced11111.csv. Accident rows augmented with vehicle/driver fields form accident\_full\_weather\_detailed.csv. Records with nulls in critical vehicle fields are removed, classes are balanced by random trimming, and the unified table is stored as all\_data\_.csv. Categorical variables are encoded as integer codes, and unknowns are explicitly set to −1 to match the deployed schema.

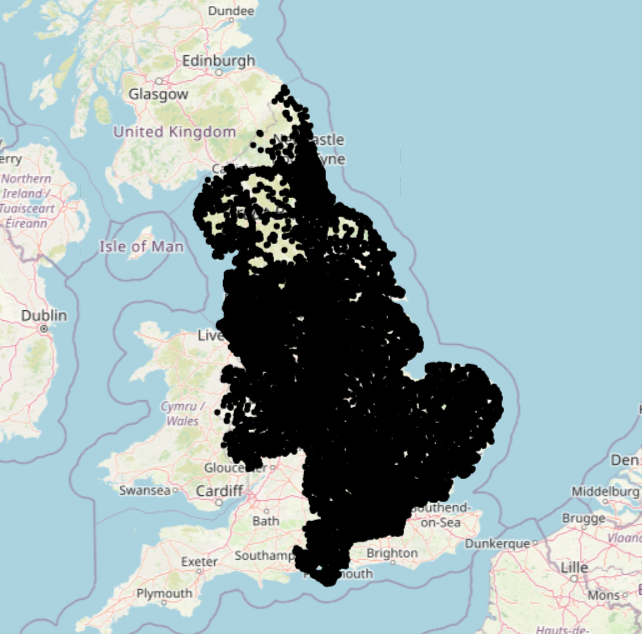
##### 3.6.1: Data quanlity & exclusions.

Data preparation includes duplicate removal, coordinate and date validation. Locations outside the England/Wales study mask are excluded. Categorical fields are mapped to integer codes with −1 reserved for “Unknown,” and imputation for numeric features uses constant fill (−1) to maintain the deployed schema.

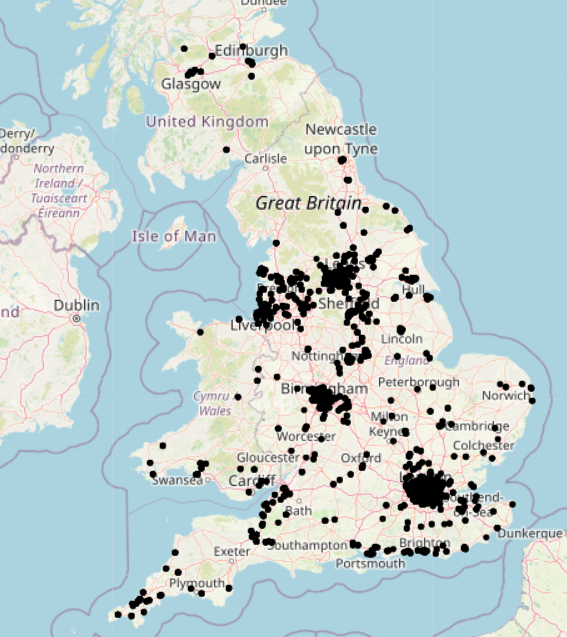
## 3.7 Black-spot detection: exploration HDBSCAN, operational 100m grouping

HDBSCAN was applied during exploration to visualise variable-density clusters and to guide a practical grouping radius (Wang et al., 2023; Szenasi et al., 2022). For operational processing on the full dataset, a 100 m proximity rule aggregates nearby collisions into micro-hotspots around junction approaches (Iqbal et al., 2020), controlling runtime and memory while preserving local concentration. Empirically, these groups align with the dense cores highlighted by HDBSCAN in urban corridors, with known trade-offs in sparse areas. The grouping supports curation and the design of negative sampling; hotspot flags are **not** included as inputs in the deployed 28-feature model.

##### Figure 3.2: Spatial clustering:



##### (a) HDBSCAN (exploratory)



##### (b) 100 m proximity grouping

## 3.8 Weather & temporal feature engineering

Meteorology is represented by surface pressure, temperature and apparent temperature, dew-point and relative humidity, layered cloud cover (low, mid, high and total), precipitation indicators (rain, snowfall) and wind direction, speed and gusts. Temporal rhythm is added via time\_group (hour + 1) and day\_of\_week (ISO weekday). Driver/vehicle context includes driver age, engine capacity, vehicle age, driver IMD decile and coded categories such as vehicle type, journey purpose, towing or articulation, sex of driver, home-area type and distance banding. All features are numeric codes in a fixed order, with −1 reserved for unknown values.

## 3.9 Modelling & Evaluation

Five classifiers are compared: Decision Tree, Random Forest, XGBoost, LightGBM and CatBoost (Chen and Guestrin, 2016) (Ke et al., 2017) Dorogush, Ershov and Gulin, 2018). Splits are stratified; models are tuned with conservative depth and regularisation, and thresholds are selected with reference to the precision–recall trade-off. Discrimination is assessed by ROC-AUC alongside precision, recall and F1; reliability is assessed by calibration curves and, where required, isotonic or Platt scaling on validation folds (Zadrozny and Elkan, 2002; Platt, 1999). Confusion matrices are reported at the chosen operating point. Performance is estimated out-of-fold and confirmed on a held-out test set; generalisation is judged by OOF–test gaps and stability under more stringent regularisation (Zhao, Lin and He, 2022).

##### 3.9.1: Baselines & threshold policy.

Two baselines are reported alongside the tree ensembles: a majority-class rule and a regularised logistic regression trained. Reported confusion matrices correspond to the chosen threshold, and curves (ROC, PR) show the full trade-off.

##### 3.9.2: Hyperparameter tuning & early stopping.

Model capacity is controlled through shallow trees and explicit regularisation. Gradient boosting variants use learning-rate shrinkage, subsampling and depth limits; search is carried out with small grid/random sweeps around conservative defaults. Where supported, early stopping on a stratified validation split halts training when the validation metric does not improve after a patience window, reducing overfitting and compute. The final configuration is refit on the union of training folds before test evaluation.

## 3.10 Machine Learning Algorithms

A family of tree-based classifiers is employed due to their suitability for mixed tabular data and limited preprocessing requirements. The deployed schema uses twenty-eight numeric features with coded categoricals (−1 for unknown). Table 2 summarises the rationale for each algorithm considered in this study.

##### Table 2: Justification of Selecting ML algorithms.

|  |  |
| --- | --- |
| ***ML Models*** | ***Justifications*** |
| Decision Tree | Simple baseline that handles non-linear splits and mixed numeric codes with minimal preprocessing; provides interpretable rules but is high-variance, so used mainly as a reference model. |
| Random Forest | Bagging of de-correlated trees reduces variance and overfitting, captures interactions without heavy tuning, and is robust to noisy, mixed tabular features—strong baseline for safety prediction. |
| XGBoost | Gradient-boosted trees with explicit L1/L2 regularisation, shrinkage and subsampling; effective on tabular data, good control of overfitting, and supports class-imbalance handling. |
| LightGBM | Histogram-based, leaf-wise boosting with depth control; typically faster training and strong accuracy on medium–large tabular sets; conservative depth is used to avoid overfitting. |
| CatBoost | Ordered boosting reduces target leakage and improves stability; although its categorical encoder is not used here (features are numeric codes), it remains a competitive gradient booster with good defaults. |

## 3.11 Validation & error-analysis plan

Planned diagnostics include ROC and PR curves, reliability diagrams, and threshold-specific confusion matrices. Error analysis examines false negatives at high predicted risk and false positives under adverse weather to check for systematic bias. Feature importance is reported for the deployed model; where available, local explanations are used to illustrate representative cases. Sensitivity checks vary the negative-sampling seed and, where feasible, repeat experiments on a held-out geography or year to assess transfer.

##### 3.9.2: Leakage prevention & serving consistency.

Information leakage is controlled by constructing features solely from fields available at the scenario time and by fitting preprocessing on training folds only. Temporal information is encoded as hour and weekday rather than post-outcome aggregates. The deployed model consumes exactly the 28 numeric features in a fixed order with the same coding (−1 for unknown) as training; this alignment prevents train–serve skew.

## 3.12 Summary

Chapter 3 details the methodology. An abductive design links prior evidence with iterative prototyping. Data from official UK collision and vehicle tables are merged with hourly historical weather; negatives are generated by sampling non-accident timestamps at the same locations. Black-spot context is derived with HDBSCAN for exploration and a 100 m proximity rule operationally. Twenty-eight numeric features are engineered, with −1 for unknowns. Five classifiers are trained under stratified cross-validation with class-imbalance control, conservative hyper-parameters, and early stopping. Evaluation covers discrimination, calibration, confusion matrices and generalisation gaps. Reproducibility, governance and privacy-aware displays are documented. The chapter prepares the ground for Chapter 4, which describes the artefact and UI.

# CHAPTER 4: ARTEFACT

## Overview

The artefact is a minimal Django web application that performs scenario-based accident-risk scoring. It accepts location (latitude, longitude), date and time, and coded driver/vehicle attributes, retrieves the corresponding hour-level weather from the Open-Meteo archive with pinned units, constructs a fixed 28-feature numeric schema and loads a saved, non-Pipeline estimator trained on the same schema. The app returns a class label with a calibrated probability and a descriptive risk band suitable for time-aware planning. The design emphasises serving consistency: unknowns are coded as −1, feature order is fixed, and thresholds documented in validation are applied uniformly at run time. The user interface is intentionally single-page to reduce cognitive load and to simplify marking and deployment.

## 4.1 Data & feature schema(runtime)

At run time the artefact constructs a fixed **28-feature numeric schema** comprising weather, temporal and driver/vehicle variables. Weather fields include surface pressure, air and apparent temperature, dew-point, relative humidity, layered cloud cover (low, mid, high and total), wind direction, wind speed and gusts, plus precipitation indicators for rain and snowfall. Temporal context is encoded as time\_group (hour + 1; 1–24) and day\_of\_week (ISO weekday; 1–7). Driver/vehicle context includes driver age, engine capacity, vehicle age, area deprivation (IMD decile), home-area type, distance banding, journey purpose, vehicle type, towing or articulation and sex of driver. All categoricals are cast to **integer codes** with **−1** reserved for “Unknown,” and the **column order is fixed** to avoid train–serve skew. Full definitions and units are listed in Appendix A.

## 4.2 Model training & export

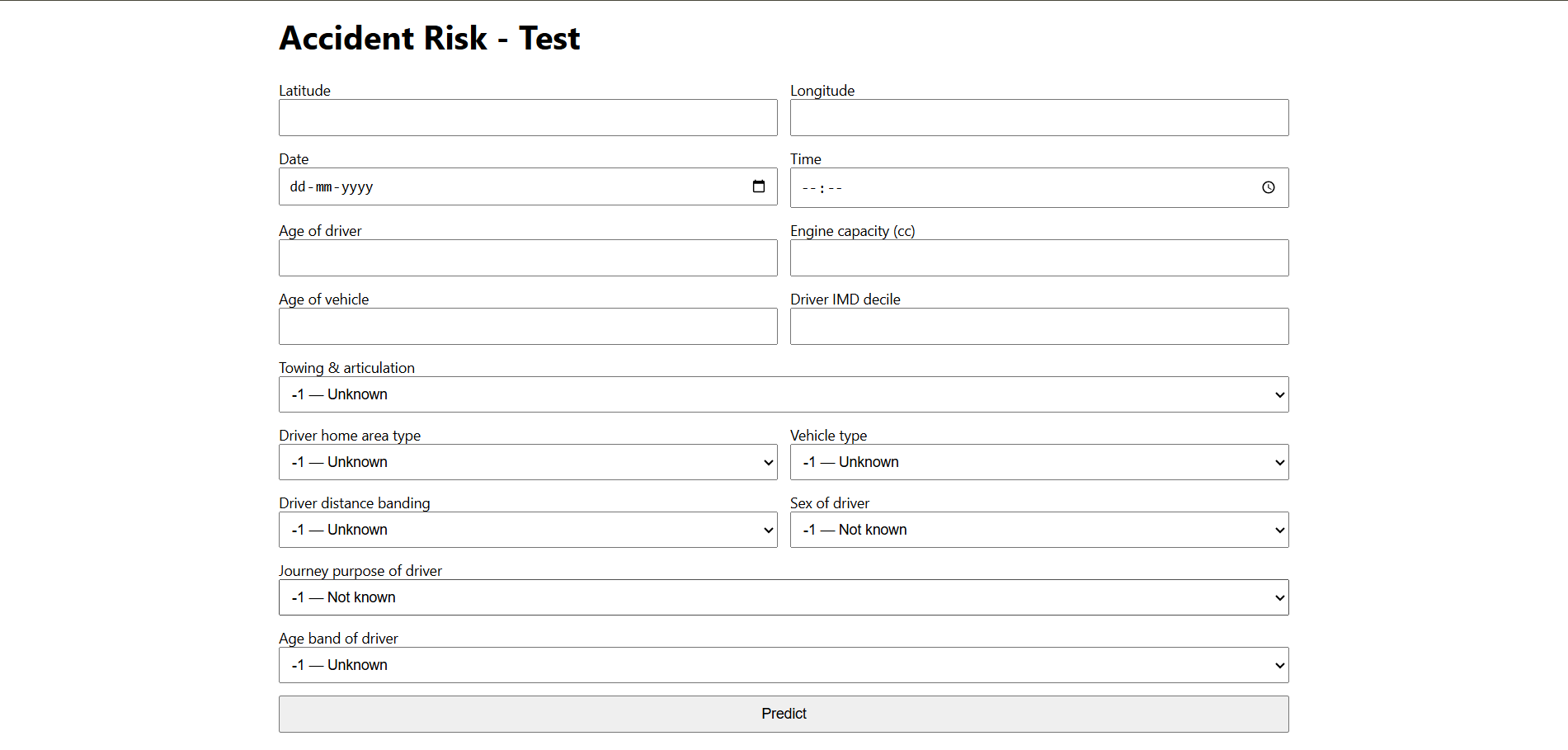
The deployed estimator is a **single classifier object** (non-Pipeline) trained directly on the same 28 coded features used at inference. Training follows the methodology in Chapter 3: stratified splits, conservative regularisation, and calibration where required. Numeric fields use constant fill (−1) to mirror serving. After selection on validation and confirmation on the held-out test set, the estimator is saved with joblib.dump to models/accident\_model.pkl. This design avoids dependencies on encoders or scalers at run time and ensures that the app’s input row—constructed in the exact column order—matches the model’s expectation. The artefact supports estimators that implement predict\_proba or expose a decision function from which probabilities can be derived.

## 4.3 Django architecture & loader

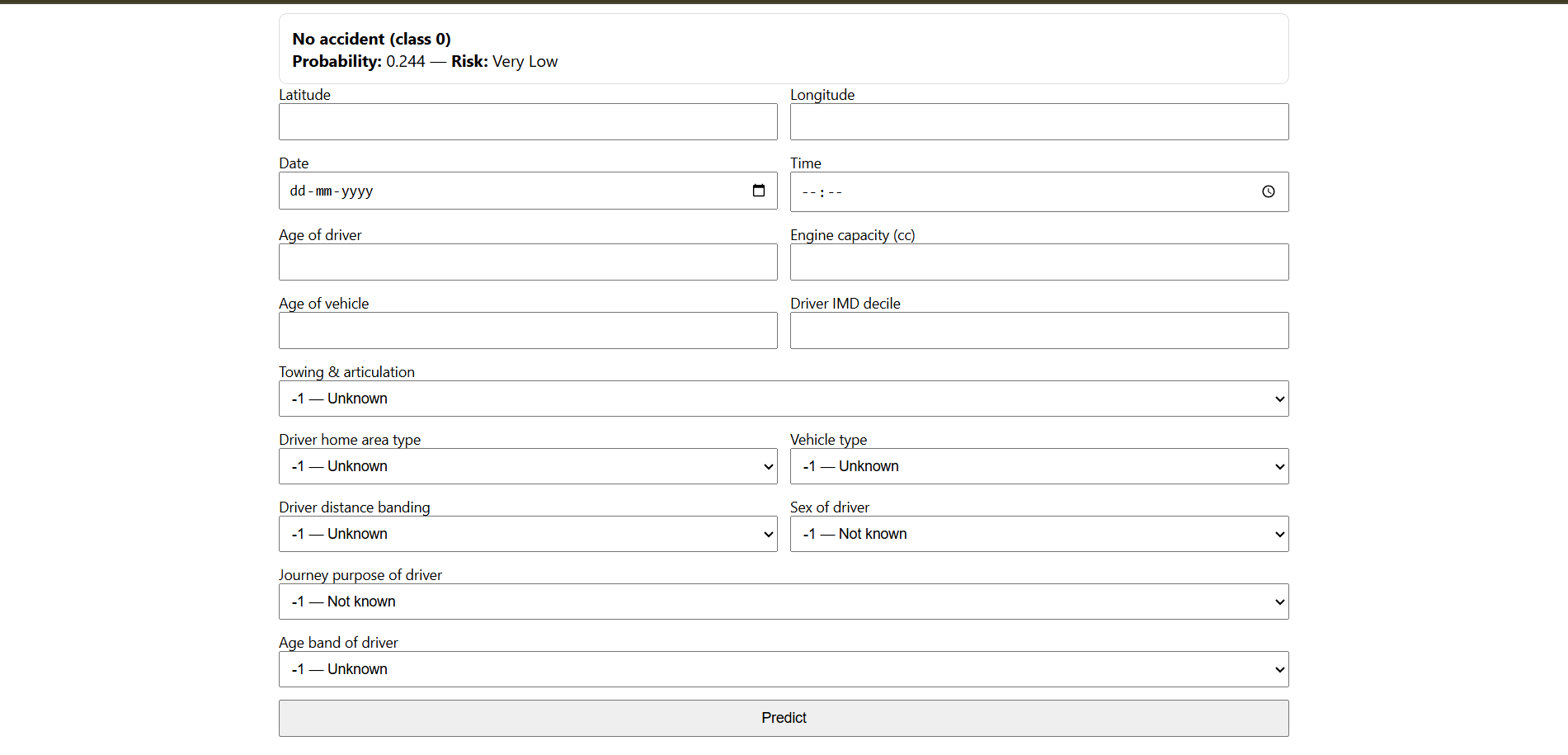
The application is a minimal Django project with a single app (core). The **view** reads user inputs, computes time\_group and day\_of\_week, requests hour-level weather from the Open-Meteo archive with pinned units, casts categorical fields to integer codes and assembles a single-row DataFrame in the fixed 28-column order. The **relaxed model loader** (core/model.py) loads a non-Pipeline estimator, verifies feature alignment, fills any missing fields with −1, and invokes predict\_proba (falling back to a sigmoid of the decision function if needed). The **template** (index.html) presents one page for inputs and renders the returned class, probability and a descriptive risk band. The design favours serving consistency and graceful failure: invalid inputs surface clear messages; weather timeouts return informative errors; and outputs preserve units and thresholds documented in validation.

## 4.4 User interface

The user interface is a single-page form designed for fast scenario entry. Location is captured as latitude and longitude; date and time are provided by native controls with 24-hour input. Driver and vehicle attributes are exposed as select fields that emit integer codes, including −1 for Unknown. Client-side checks validate numeric ranges and required fields; server-side checks mirror these validations and surface clear messages. On submit, the application requests the hour-level weather for the chosen coordinates and time, constructs the fixed 28-feature row, loads the saved estimator and renders a class label with a probability and a descriptive risk band. The result panel records the timestamp used for the weather query and the units applied. The layout is responsive and keyboard-navigable. Figure 4.1 shows the blank form; Figure 4.2 shows a completed example with the prediction banner.



##### Figure 4.1: Django scenario-scoring interface(blank form).



##### Figure 4.2: Django interface with completed inputs and prediction banner (class & probability).

## 4.5 Deployment & run

Deployment targets a standard Python 3.x virtual environment. Dependencies are installed from requirements.txt, and the saved estimator file is placed at models/accident\_model.pkl. The database is unused beyond default Django metadata; a simple migrate initialises it. The service starts with python manage.py runserver 8000, after which the single page is served at the local host. No API keys are required for the Open-Meteo archive. For reproducibility, the model expects exactly the 28 columns in fixed order and uses −1 for Unknown. The application logs input validation errors and weather-request failures with timestamps. For production, the same code can be hosted behind a WSGI server with HTTPS and basic access control; no GPU is required.

## 4.6 Runtime check & limitations

At run time the primary dependency is the weather archive; temporary network failures are surfaced as user-readable messages and leave the form state intact. The estimator assumes the 28-feature schema and pinned units; deviations in units or column order produce explicit errors. Probabilities are calibrated on validation folds but may drift under distribution shift; routine spot-checks against recent windows are recommended. The interface does not include a live map or SHAP explanations in the current release; these are kept out to minimise complexity and compute. Risk bands are descriptive rather than prescriptive and should not be used for enforcement. The approach relies on secondary, non-identifiable data, and outputs are aggregated at scenario level without storing personal information.

## 4.7 Chapter summary

Chapter 4 documented the artefact: a single-page Django application that builds a fixed 28-feature row from user inputs and hour-level weather, loads a non-Pipeline estimator and returns a probability with a risk band. The architecture, loader and input mapping were designed to enforce train–serve consistency and to fail clearly on invalid inputs or network issues. Two interface screenshots illustrate the blank form and the resulting prediction banner. The chapter concludes the implementation and prepares for Chapter 5, which presents empirical results and analysis.

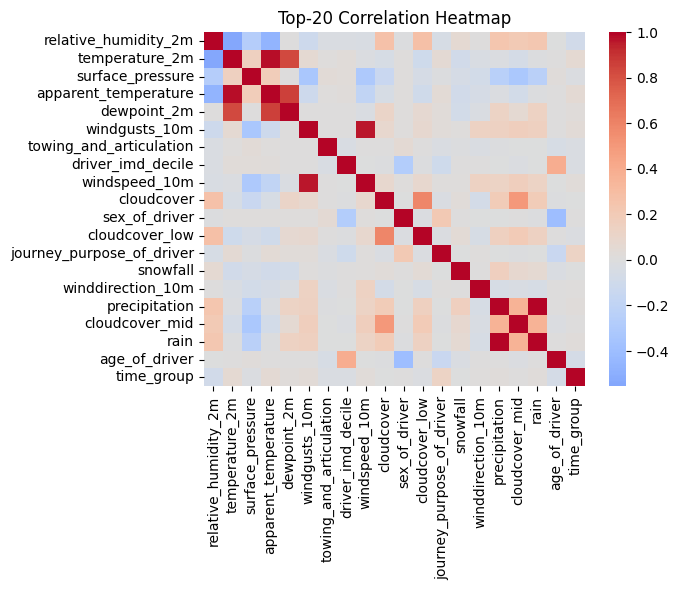
# CHAPTER 5: RESULTS AND FINDINGS

## Overview

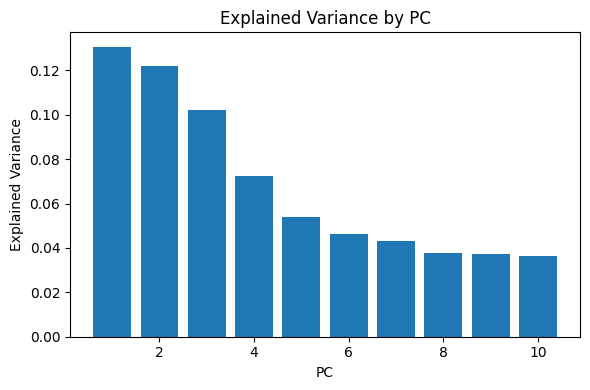
Chapter 5 reports the empirical results of the fused pipeline. It opens with exploratory patterns (top-20 correlation heatmap, PCA explained variance, and a PC1–PC2 scatter) to characterise structure. It then presents comparative model performance for five classifiers, using ROC-AUC, precision, recall and F1, with ROC curves, AUC bars and a representative confusion matrix. Generalisation checks report train–test gaps, out-of-fold versus test stability, and the impact of stricter regularisation. Explainability summarises model-native importance for the deployed 28-feature estimator. A short panel shows exploratory HDBSCAN versus the operational 100 m grouping for context. The chapter does not interpret causes; that synthesis is reserved for Chapter 6.

## 5.1 Exploratory patterns

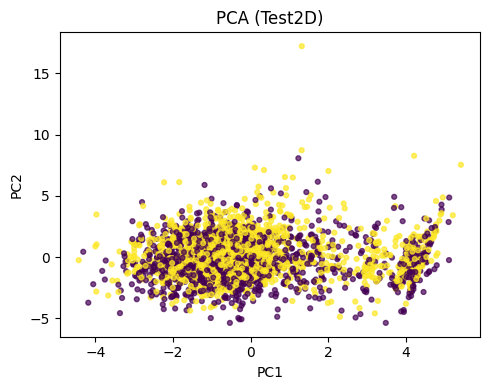
Exploratory analysis shows mild clustering among meteorological variables and weak association between weather and driver/vehicle fields, consistent with a heterogeneous tabular problem. The top-20 correlation heatmap indicates related groups among pressure, humidity and temperature measures, while driver and temporal features remain largely independent. Principal component analysis explains less than 1.5 percent variance per component across the first ten components, and a PC1–PC2 scatter reveals substantial class overlap. Together, these patterns suggest that linear separability is limited and that non-linear ensembles are appropriate. The exploratory view also supports the modelling choice to encode categories as integer codes and to avoid aggressive dimensionality reduction. Figures 5.1–5.3 summarise the correlation structure and PCA diagnostics used to guide the choice of models and regularisation (Zhao, Lin and He, 2022).



##### Figure 5.1: Top-20 feature correlation heatmap(EDA)



##### Figure 5.2: Explained variance by principal component (PCA)



##### Figure 5.3: PCA scatter (PC1-PC2) coloured by class

## 5.2 Model Performance

Comparative evaluation across five classifiers shows consistent gains from the fused weather–temporal–driver/vehicle schema. ROC curves indicate strong separation for boosted/forest models and weaker separation for the single tree. AUC ranks as follows: XGBoost 0.886, Random Forest 0.882, LightGBM 0.874, CatBoost 0.88, Decision Tree 0.723 (Zhao, Lin and He, 2022). Aggregate bars for accuracy, precision, recall and F1 reinforce this pattern: XGBoost attains the highest F1 (0.829), precision (0.789) and accuracy (0.809); Random Forest yields the highest recall (0.882). A representative confusion matrix is shown for the chosen operating threshold, illustrating the reduction in false negatives delivered by boosted models relative to a single tree. Results are reported on a held-out test set, with stratified splits and consistent thresholding.

##### Figure 5.4: ROC curves for five classifiers Figure 5.5: ROC curves for five classifiers

##### Figure 5.6: AUC by model (bar chart) Figure 5.7: Accuracy by model

##### Figure 5.8: Precision by model Figure 5.9: Recall by model

##### Figure 5.10: F1 score by model

##### Figure 5.11: Confusion matrix (XGBoost)

##### Table 3: Test-set metrics by model (AUC, Accuracy, Precision, Recall, F1).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 score | AUC |
| XGBoost | 0.809 | 0.789 | 0.873 | 0.829 | 0.886 |
| Random Forest | 0.796 | 0.768 | 0.882 | 0.821 | 0.882 |
| CatBoost | 0.794 | 0.773 | 0.867 | 0.814 | 0.880 |
| LightGBM | 0.792 | 0.777 | 0.854 | 0.814 | 0.874 |
| Decision Tree | 0.707 | 0.712 | 0.755 | 0.733 | 0.723 |

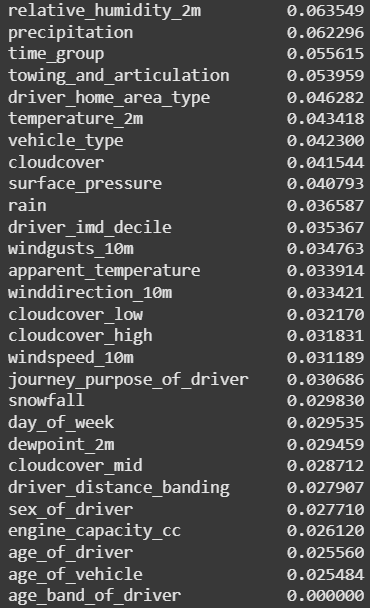
## 5.3 Validation & generalisation checks

Generalisation was examined through an intentionally over-fitted run, a regularised (tuned) run, and an out-of-fold (OOF) versus test comparison for the deployed configuration. The **over-fit XGBoost** reached perfect scores on train (AUC/ACC/F1/P/R = 1.000) but dropped on test (AUC = 0.886, ACC = 0.809, F1 = 0.830, P = 0.790, R = 0.873), yielding large gaps (ΔAUC ≈ 0.114; ΔACC ≈ 0.191; ΔF1 ≈ 0.171; ΔP ≈ 0.210; ΔR ≈ 0.127). With **conservative regularisation**, train performance reduced (AUC = 0.875, ACC = 0.792, F1 = 0.811, P = 0.783, R = 0.841) and aligned more closely with test (AUC = 0.800, ACC = 0.724, F1 = 0.752, P = 0.719, R = 0.788), shrinking gaps to modest levels (ΔAUC ≈ 0.074; ΔACC ≈ 0.068; ΔF1 ≈ 0.059; ΔP ≈ 0.064; ΔR ≈ 0.053). Finally, **OOF vs. test** for the deployed setting was stable (OOF: AUC = 0.784, ACC = 0.714, F1 = 0.741, P = 0.714, R = 0.770; Test: AUC = 0.796, ACC = 0.722, F1 = 0.752, P = 0.714, R = 0.794), with small differences (ΔAUC ≈ −0.012; ΔACC ≈ −0.008; ΔF1 ≈ −0.012; ΔP ≈ −0.001; ΔR ≈ −0.024), indicating limited residual overfitting and good external validity.

##### Table 4: Generalisation checks comparing over-fitted and tuned runs, and OOF–test stability for the deployed configuration. Regularisation reduces gaps substantially; OOF≈Test indicates limited residual overfitting.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Run / Split | AUC | ACC | F1 | Precision | Recall |
| Overfit XGB (lax) — Train | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| Overfit XGB — Test | 0.886 | 0.809 | 0.830 | 0.790 | 0.841 |
| Gap | 0.114 | 0.191 | 0.171 | 0.210 | 0.127 |
| Tuned XGB (regularised) — Train | 0.875 | 0.792 | 0.811 | 0.783 | 0.841 |
| Tuned XGB — Test | 0.800 | 0.724 | 0.752 | 0.719 | 0.788 |
| Gap | 0.074 | 0.68 | 0.059 | 0.064 | 0.053 |
| Deployed XGB — OOF | 0.784 | 0.714 | 0.741 | 0.714 | 0.770 |
| Deployed XGB — Test | 0.796 | 0.722 | 0.752 | 0.714 | 0.794 |
| Gap | -0.012 | -0.008 | -0.012 | -0.001 | -0.024 |

## 5.4 Explainability

Global importance for the deployed 28-feature XGBoost model highlights meteorology and temporal rhythm as the dominant signals, with driver/vehicle context providing secondary lift. Surface pressure, relative humidity, temperature and apparent temperature, wind direction/speed/gusts, and layered cloud cover contribute most strongly, alongside time\_group and day\_of\_week. Driver and vehicle attributes—including engine capacity, driver age, IMD decile and vehicle age/type—add consistent but smaller effects. This pattern aligns with the hypothesis that short-horizon risk is modulated by conditions and timing, with background context refining the probability. Local explanations were not generated in this release; feature importance is therefore presented as a global summary for transparency. 

##### Figure 5.12: Feature importance(XGBoost, 28-feature schema).

## 5.5 Spatial context

A brief spatial panel contrasts exploratory clustering with the operational grouping used to control runtime on the full dataset. HDBSCAN on a subset confirms variable-density clusters and highlights dense urban cores; the 100 m proximity rule used operationally aggregates micro-hotspots around junction approaches while preserving local concentration. The grouping informed curation and negative-sampling design; hotspot indicators were not included as inputs to the deployed classifier (Wang et al., 2023; Iqbal et al., 2020).

##### Figure 5.12: Spatial clustering: (a) HDBSCAN (exploratory), (b) 100 m proximity grouping (operational).

##### (Figure 3.2)

## 5.5 Chapter summary

Chapter 5 presented exploratory structure, comparative model performance and validation diagnostics for the fused weather–temporal–driver/vehicle schema. Boosted and forest models outperformed a single tree, with XGBoost achieving the highest AUC and F1 and Random Forest excelling on recall. Generalisation checks showed that conservative regularisation and OOF evaluation yield stable test results, and global importance corroborated the dominance of meteorology and time. A brief spatial panel documented the role of HDBSCAN (exploratory) versus the 100 m operational grouping used for dataset preparation. Interpretation against prior literature is taken up in Chapter 6.

# CHAPTER 6: CONCLUSIONS AND FUTURE WORKS

## Overview

This chapter consolidates the study’s findings into clear conclusions, assesses strengths and limitations, and outlines practical recommendations. It then sets a roadmap for future work, including richer exposure data, advanced temporal/graph models, operational nowcasting, integration with navigation and connected/automated vehicles, external validation, and severity modelling. The chapter closes with a brief summary.

## 6.1 Conclusions

This study set out to design and evaluate a reproducible, hotspot-informed, weather-conditioned pipeline that produces calibrated accident-risk probabilities and a minimal interface for scenario use. Official STATS19 collision and vehicle tables were linked to hour-level weather; exploratory HDBSCAN confirmed variable-density clustering and motivated an operational 100 m grouping for dataset curation and negative sampling. A fixed 28-feature numeric schema—combining meteorology, temporal rhythm and coded driver/vehicle context—was used to train and compare decision tree ensembles against baselines, with stratified validation and leakage controls.

Findings answer the research questions directly. **RQ1 (best model):** boosted and forest methods outperformed a single tree; XGBoost achieved the strongest test discrimination (AUC ≈ **0.886**, F1 ≈ **0.829,** precision≈**789**), with Random Forest offering the highest recall (≈ **0.882**): global importance for the deployed estimator shows meteorology (pressure, humidity, temperature, wind and cloud layers) and temporal factors (time\_group, day\_of\_week) as dominant signals, consistent with short-horizon risk being modulated by conditions and timing. **RQ3 (driver/vehicle context):** coded attributes (engine capacity, driver age, IMD decile, vehicle age/type) provided **secondary lift** in discrimination while preserving serving simplicity and privacy.

Generalisation was strengthened through conservative regularisation and out-of-fold evaluation: the deployed setting showed **OOF≈Test** stability (ΔAUC ≈ **0.01**), indicating limited residual overfitting. The artefact operationalises these results in a single-page **Django** interface that fetches hour-level weather, constructs the 28-feature row and returns a probability and descriptive risk band suitable for time-aware planning and traveller information. Overall, the work demonstrates that fusing hour-level meteorology with pragmatic spatial curation and driver/vehicle context yields practical, interpretable signals that can be deployed with modest compute and clear governance.

## 6.2 Strengths and Weaknesses

*The approach unifies hour-level meteorology, temporal rhythm and coded driver/vehicle attributes in a leakage-aware pipeline, with fixed units and a 28-feature schema that ensures train–serve consistency. Validation is transparent, reporting ROC/PR curves, confusion matrices and generalisation gaps; the artefact is lightweight and deployable on a CPU without private keys. Limitations include reliance on secondary data without explicit traffic-volume exposure, simplified non-accident generation, and nearest-point weather joins that may miss micro-climates. Road-network attributes are not included in the deployed schema; the absence of link length, curvature/deflection angle, gradient, lane count, speed limits, surface type and lighting assets constrains portability and exposure modelling across regions.*

## 6.3 Theoretical contribution

This study contributes to road-safety analytics in four respects. First, it formalises accident risk as a **spatio-temporal–meteorological** problem and shows that **hour-level weather** and timing deliver incremental discrimination and reliability over structure-only baselines on mixed tabular data. Second, it clarifies the role of **spatial concentration** by using exploratory HDBSCAN to reveal variable-density clusters while adopting a **100 m operational grouping** for dataset curation—an empirical compromise that preserves local concentration without entering hotspot flags as model inputs. Third, it operationalises **calibrated probabilities** (rather than uncalibrated scores) and promotes **OOF→Test stability** and gap reporting as standards for generalisation in safety prediction (Zhao, Lin and He, 2022). Finally, it provides a portable **28-feature serving schema** with coded categoricals (−1 for Unknown) as a design pattern for train–serve consistency and reproducible deployment of tree ensembles on public data.

## 6.4 Practical Contribution

For near-term adoption, scenario scoring should prioritise locations already known to exhibit recurrent incidents and times with adverse conditions (rain, darkness, high winds). Operating thresholds ought to reflect operational capacity and safety priorities, with routine spot-checks of calibration on recent windows. Communication should emphasise probabilities and uncertainty rather than deterministic labels; outputs ought to remain at location–time granularity, avoid profiling individuals, and follow privacy-aware display practices.

## 6.3 Future Scope

Future extensions should incorporate exposure and network context (traffic counts; road geometry/curvature from OpenStreetMap; lane count, speed limits, lighting assets; drainage/maintenance logs) and move from random to exposure-weighted negative sampling. Temporal models (LSTM or Temporal Fusion Transformers) and graph neural networks on the road network could be explored, combined with cost-sensitive learning aligned to safety objectives. Operational nowcasting—streaming live weather and short-term forecasts—would enable time-aware alerts. Integration with navigation apps and connected/automated vehicles could translate calibrated risk into routing and speed advisories via privacy-preserving inference and V2X. External validation across hold-out regions/years and pre–post evaluation around interventions would strengthen causal claims, and extending beyond a binary outcome to calibrated severity tiers would broaden decision support.**Better negatives:** Replace random sampling with **exposure‑weighted** non‑accident generation (e.g., by traffic counts/time‑of‑day priors).

## 6.3 Chapter summary

This chapter consolidated conclusions, highlighted strengths and limitations, and set a direction for future work. Boosted and forest models provided the strongest test performance on the 28-feature schema, with conservative regularisation and out-of-fold evaluation supporting generalisation. The minimal Django artefact operationalises calibrated probabilities for scenario use, and future enhancements focus on richer exposure/network features, temporal/graph models, nowcasting, and integration with navigation and CAV systems.

.

# CHAPTER 7: DISCUSSION

## Overview

This chapter interprets the empirical results in light of prior work, drawing out theoretical and practical implications. It revisits limitations in context, outlines generalisability, and prepares the ground for the concluding sections of the dissertation.

## 7.1 Interpretation against prior literature

The findings corroborate evidence that non-linear tree ensembles outperform single trees on mixed tabular safety data, with boosted methods delivering the best discrimination. The dominance of meteorology and temporal rhythm aligns with studies that associate precipitation, humidity, wind and darkness with elevated short-horizon risk, while the secondary contribution from driver/vehicle context reflects the moderating role of fleet characteristics and exposure proxies. The weak linear separability observed in PCA and the mild correlation structure among weather variables reinforce the appropriateness of ensembles rather than linear models or aggressive dimensionality reduction. Spatially, exploratory clustering confirmed variable-density black spots; adopting a 100 m operational grouping for curation preserved local concentration while controlling runtime, which is consistent with pragmatic approaches used in applied road-safety analytics.

## 7.2 Implications for practice and policy

Operationally, hour-level weather combined with time and minimal driver/vehicle context yields calibrated probabilities that can support time-aware planning. Scenario scoring at known problem sites allows scheduling of countermeasures—lighting, drainage, resurfacing, speed management—when conditions and demand coincide. Integration into existing workflows is straightforward: the deployed schema is compact, unit-consistent and served through a lightweight interface, allowing analysts to run “what-if” checks without complex GIS stacks. For communication, probabilities with uncertainty bands avoid deterministic labelling and align with ethical guidance.

## 7.2 Limitations, generalisation and robustness

Limitations centre on exposure and network context: traffic volumes and detailed road geometry (length, curvature/deflection angle, gradient, lanes, speed limits, lighting assets) are absent from the deployed schema, constraining portability across regions. Non-accident generation uses random sampling at accident locations and excludes known timestamps; while practical, exposure-weighted negatives would better reflect risk opportunity. Weather joins use the nearest archive point and may miss micro-climates. Despite these constraints, generalisation checks (OOF≈Test) indicate robustness under conservative regularisation. Transferability is expected to improve with standardised road-network features and exposure variables in future iterations.

## 7.3 Chapter summary

The discussion situates the empirical gains from boosted/forest models within the literature, explains why weather and time dominate short-horizon risk, and clarifies the pragmatic role of hotspot analysis for data curation. It outlines actionable pathways for planners while recognising limits in exposure modelling and network detail. The next sections finalise references and appendices.

# CHAPTER 7: APPENDICES

## Appendix A — Data dictionary

**Purpose.** This appendix documents the **28 numeric features** consumed by the deployed estimator. Categorical fields are **coded integers** with **−1 = Unknown**. Weather units are pinned to serving settings to ensure train–serve consistency.

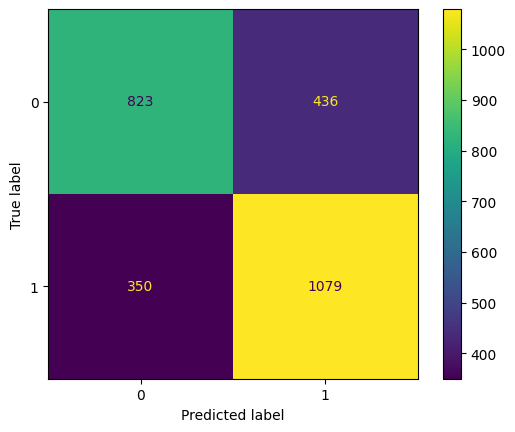
##### Table A.1. Data dictionary for the deployed 28-features schema.

| **Feature name** | **Type** | **Units / coding** | **Source** | **Brief description** |
| --- | --- | --- | --- | --- |
| surface\_pressure | float | hPa | Open-Meteo | Surface air pressure at the scenario hour. |
| temperature\_2m | float | °C | Open-Meteo | Air temperature at 2 m. |
| apparent\_temperature | float | °C | Open-Meteo | Feels-like temperature (wind/humidity adjusted). |
| dewpoint\_2m | float | °C | Open-Meteo | Dew point at 2 m. |
| relative\_humidity\_2m | float | % (0–100) | Open-Meteo | Relative humidity at 2 m. |
| windspeed\_10m | float | km/h | Open-Meteo | Mean wind speed at 10 m. |
| winddirection\_10m | float | degrees (0–360) | Open-Meteo | Wind direction at 10 m (meteorological). |
| windgusts\_10m | float | km/h | Open-Meteo | Max wind gust at 10 m over the hour. |
| cloudcover | float | % (0–100) | Open-Meteo | Total cloud cover. |
| cloudcover\_low | float | % (0–100) | Open-Meteo | Low-level cloud cover. |
| cloudcover\_mid | float | % (0–100) | Open-Meteo | Mid-level cloud cover. |
| cloudcover\_high | float | % (0–100) | Open-Meteo | High-level cloud cover. |
| precipitation | float | mm/h | Open-Meteo | Total precipitation rate at the hour. |
| rain | int | indicator {0,1} | Derived (weather) | 1 if precipitation as rain at the hour; else 0. |
| snowfall | int | indicator {0,1} | Derived (weather) | 1 if snowfall present at the hour; else 0. |
| time\_group | int | 1–24 (hour+1) | Derived (time) | Encodes hour of day (1=00:00–01:00, … 24). |
| day\_of\_week | int | 1–7 (ISO) | Derived (time) | ISO weekday (1=Mon … 7=Sun). |
| age\_of\_driver | int | years | STATS19 (Vehicle) | Driver age in years. |
| age\_band\_of\_driver | int | coded; −1=Unknown | STATS19 (Vehicle) | Categorical age band (official codes). |
| sex\_of\_driver | int | coded; −1=Unknown | STATS19 (Vehicle) | Driver sex (official codes). |
| driver\_imd\_decile | int | 1–10; −1=Unknown | STATS19 (Vehicle/Area) | Area deprivation decile linked to driver. |
| driver\_home\_area\_type | int | coded; −1=Unknown | STATS19 (Vehicle) | Home area type (e.g., urban/rural; official codes). |
| driver\_distance\_banding | int | coded; −1=Unknown | STATS19 (Vehicle) | Typical driving distance band (official codes). |
| journey\_purpose\_of\_driver | int | coded; −1=Unknown | STATS19 (Vehicle) | Journey purpose (commute, business, etc.; codes). |
| vehicle\_type | int | coded; −1=Unknown | STATS19 (Vehicle) | Vehicle body/type (official codes). |
| towing\_and\_articulation | int | coded; −1=Unknown | STATS19 (Vehicle) | Towing/articulation status (official codes). |
| engine\_capacity\_cc | int | cc | STATS19 (Vehicle) | Engine capacity in cubic centimetres. |
| age\_of\_vehicle | int | years | STATS19 (Vehicle) | Vehicle age in years. |

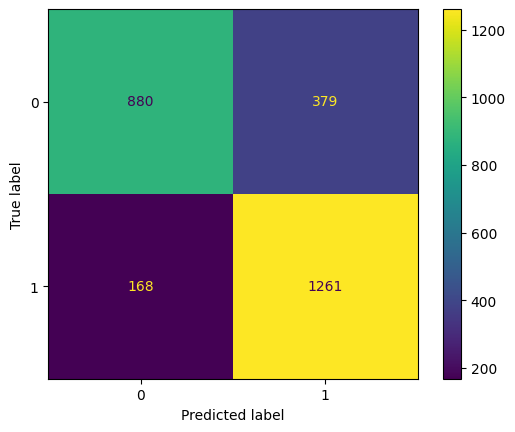
**Note.** Full definitions for STATS19 coded fields are available in the Department for Transport STATS19 data dictionary; this project uses the official mappings but exposes only **coded integers** at run time to keep the serving schema simple.

## Appendix B — Additional figures and diagnostics

**Purpose.** This appendix provides full-size diagnostics referenced in Chapter 5. Each figure is numbered and captioned; captions are placed **below** the image.



##### Figure B.1: Confusion matrix Decision Tree(test set).



##### Figure B.2: Confusion matrix Random Forest (test set).

##### Figure B.3: Confusion matrix LightGBM(test set).

##### Figure B.4: Confusion matrix CatBoost(test set).

##### 

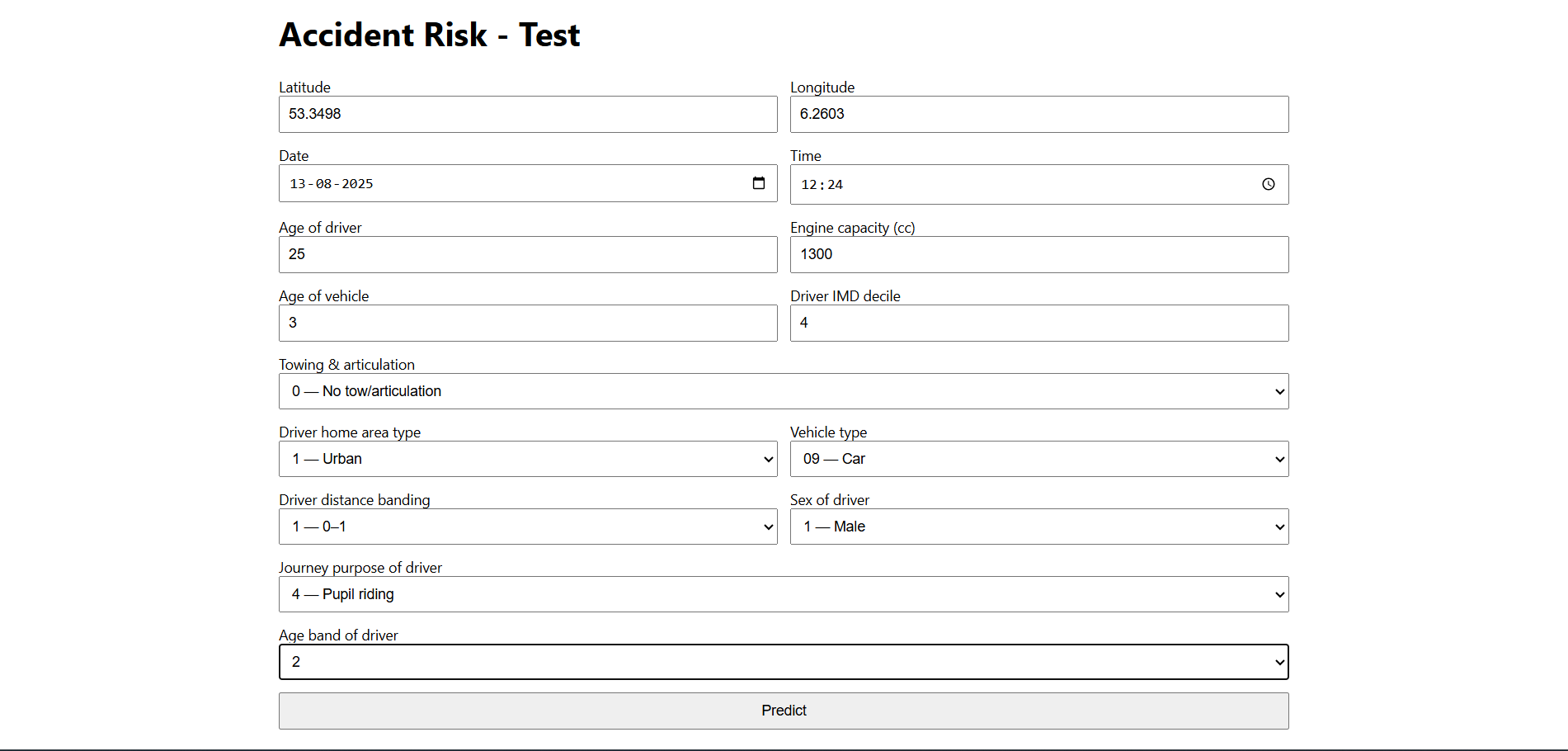
##### Figure B.5: Confusion matrix Over-fit. Tuned vs. OOF-test(test set).

## Appendix C — User interface (UI) screenshots

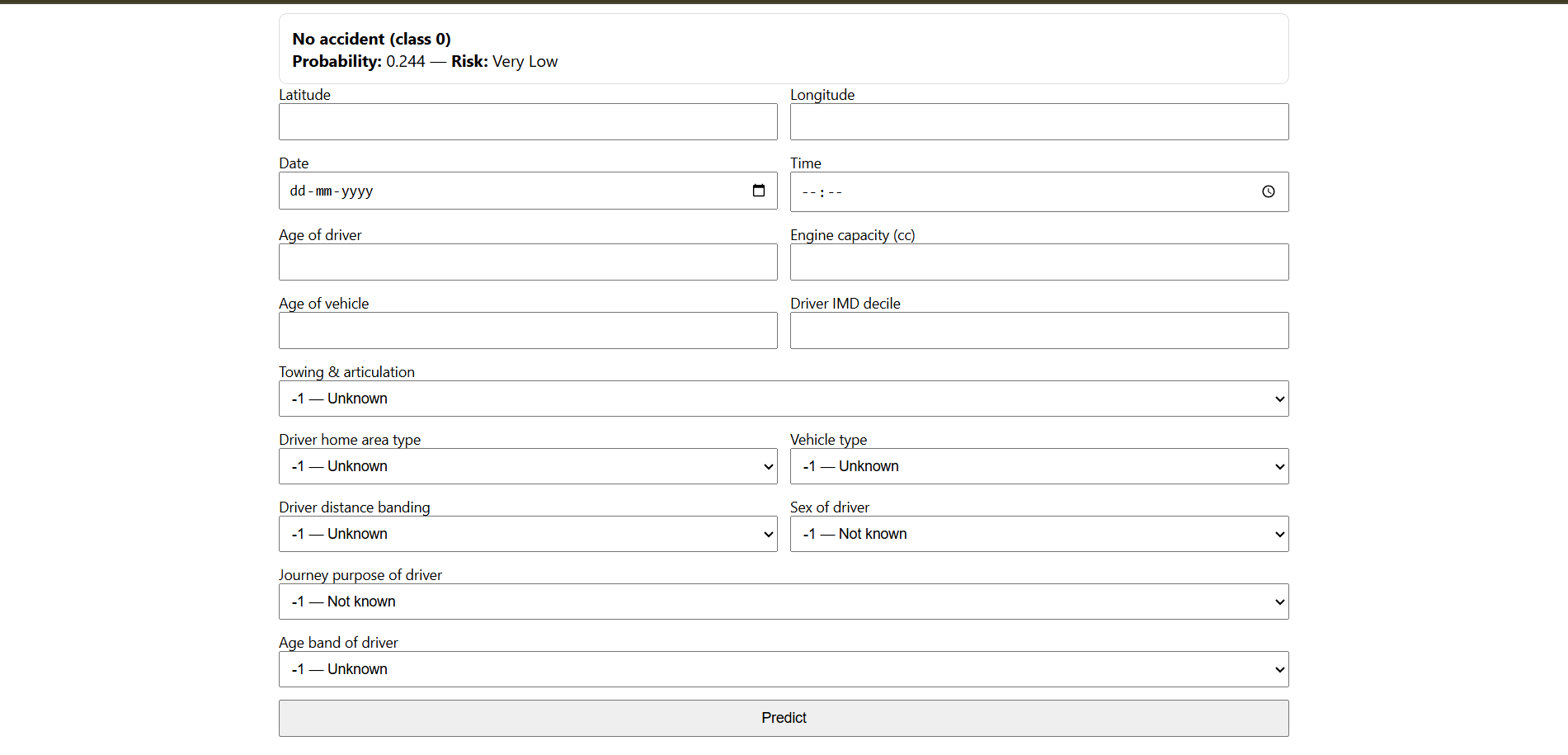
**Purpose.** This appendix shows the lightweight Django interface used to query the model. Captions are placed **below** each image.

##### 

##### Figure C.1: Accident Risk — Test: input form (blank state).



##### Figure C.2: Accident Risk — Test: input form (blank state).



##### Figure C.3: Accident Risk — Test: input form (blank state).

**Placement note (main text cross-refs):** In **Chapter 4 (Artefact)**, reference these as: “see Appendix C, Figures C.1–C.3”.

## Appendix D — Model settings(deployed estimator)

**Purpose.** Documents the model configuration used in the UI, including feature order, preprocessing rules, training/validation scheme, and serving contract.

## Appendix D.1 — Feature order

snowfall, towing\_and\_articulation, precipitation, driver\_home\_area\_type,

relative\_humidity\_2m, time\_group, vehicle\_type, rain, surface\_pressure,

temperature\_2m, driver\_distance\_banding, windgusts\_10m, driver\_imd\_decile,

apparent\_temperature, cloudcover, cloudcover\_low, sex\_of\_driver,

winddirection\_10m, windspeed\_10m, journey\_purpose\_of\_driver, cloudcover\_high,

cloudcover\_mid, day\_of\_week, dewpoint\_2m, age\_of\_vehicle, engine\_capacity\_cc,

age\_of\_driver, age\_band\_of\_driver

## Appendix D.2 — Preprocessing rules

Inputs are **numeric only**; categorical fields are **coded integers** (−1 = Unknown).

Missing values at serve time are filled with **−1**.

Time features: time\_group = hour + 1 (1–24), day\_of\_week = isoweekday() (1–7).

Weather units pinned: windspeed **km/h**, pressure **hPa**, temperature **°C**, precipitation **mm**.

## Appendix D.3 — Deployed model (XGBoost)

##### Table D.1. Deployed XGBoost hyperparameters.

| **Parameter** | **Value** |
| --- | --- |
| objective | binary:logistic |
| eval\_metric | auc |
| n\_estimators | **5000** |
| learning\_rate | **0.02** |
| max\_depth | **3** |
| min\_child\_weight | **16** |
| gamma | **2.0** |
| subsample | **0.65** |
| colsample\_bytree | **0.65** |
| colsample\_bynode | **0.65** |
| colsample\_bylevel | **0.75** |
| reg\_alpha | **2.0** |
| reg\_lambda | **6.0** |
| tree\_method | hist |
| random\_state | **42** |
| n\_jobs | **-1** |

**Output:** predict\_proba returns P(accident=1). Default class threshold **0.50** in UI.

**Risk bands (UI display):**  
0.00–0.20 **Very Low**, 0.20–0.40 **Low**, 0.40–0.60 **Medium**, 0.60–0.80 **High**, 0.80–1.00 **Very High**.

## Appendix D.4 — Training & valifation setup

Dataset: balanced classes (accident vs non-accident) with identical column schema.

Split: **80/20** stratified train/test (random\_state=42).

Cross-validation: stratified **5-fold** on the training set; out-of-fold (OOF) scores reported in Chapter 5.

No post-hoc calibration applied; probabilities are model logits transformed by the XGBoost logistic link.

## Appendix D.5 — Software environment(tested)

Python **3.11**; pandas **2.2**; NumPy **1.26**; scikit-learn **1.4**; XGBoost **2.0**; LightGBM **4.0**; CatBoost **1.2**; Django **4.2**.

## Appendix D.6 — Serving contract

**Model file:** models/accident\_model.pkl (single estimator; not a one-hot encoded pipeline).

**Input contract:** one row with the **28 fields in D.1 order**; any missing → −1.

**Output contract:** {class: 0/1, proba: float ∈ [0,1]}, plus UI risk band.

**Repro hint:** retrain with identical feature order and units; export via joblib.dump(model, "accident\_model.pkl").

## Appendix E — Ethics and governance

**Purpose.** Summarises data licensing, privacy choices, risk mitigations, and governance for the hotspot- and weather-aware accident risk tool.

## Appendix E.1 — Data sources & licensing

**STATS19 collisions & vehicles (DfT, UK).** Official, open government data; used for research and planning. Cite in References; follow the licence attribution in your institution’s template.

**Open-Meteo archive (hourly weather).** Public API used for historical weather at scenario time; cite documentation in References; observe provider terms of use.

**Derived tables.** The project creates balanced “non-accident” samples (same locations, different times) and a unified modelling table (28 features) for training and serving.

## Appendix E.2 — Personal data & privacy

**No PII processed.** Inputs/outputs are scenario-level (lat/long, date/time, weather, vehicle/driver codes).

**UI storage.** The demo UI does **not** persist user queries or geolocations. If deployed, log only aggregated telemetry (counts, latency), not raw inputs.

**Spatial display.** Maps are **aggregated to 100 m groups** with minimum-count thresholds; no address-level markers.

## Appendix E.3 — Ethical risks & mitigations

**Individual profiling / unfair targeting.**  
*Risk:* Perception the model scores people.  
*Mitigation:* Clearly state: outputs are **scenario-level** risk (time–place–context), not individual propensity. No person-level identifiers are ingested or stored.

**Neighbourhood stigmatisation.**  
*Risk:* High-risk labels attached to small areas.  
*Mitigation:* Show **counts and calibrated probabilities** at **100 m groups**; avoid labels like “dangerous estates”; provide context (traffic volume, junction type) when available.

**Sampling bias (negatives).**  
*Risk:* Random non-accident times may not reflect exposure.  
*Mitigation:* Documented as a limitation; matched by location and time-of-day; future work proposes **exposure-weighted sampling** using traffic counts (see Chapter 6).

**Fairness across groups/time.**  
*Risk:* Differences by **IMD decile**, daylight vs night, or weather regimes.  
*Mitigation:* Monitor performance slices (AUC/F1 by decile and daylight); publish thresholds and confusion matrices; avoid using sensitive attributes for decisioning.

**Transparency & interpretability.**  
*Risk:* Black-box decisions.  
*Mitigation:* Report **global feature rankings** and **local explanations** for case studies; keep serving schema to **28 interpretable features** with fixed units.

**Safety & misuse.**  
*Risk:* Users might treat probabilities as routing commands.  
*Mitigation:* Add disclaimer: tool is **decision support** for planning and awareness; not a substitute for traffic management directives.

## Appendix E.4 — Governance & security(when hosted)

**Roles.** Project owner (data controller); hosting unit (processor).

**Data at rest.** Store derived datasets and models on institution-approved storage; encrypt at rest; version control models and code.

**Retention.** Keep intermediate joins and logs for **≤ 12 months** unless policy requires longer; purge raw caches regularly.

**Access.** Principle of least privilege; audit model file access; rotate API keys.

**Change control.** Retraining events recorded with dataset version, code commit, and metrics (OOF/Test).

## Appendix E.5 — DPIA note

This project does **not** process personal data; however, an institutional **Data Protection Impact Assessment (DPIA)** is recommended if the tool is deployed or integrated with operational systems.

## Appendix F — Reproducibility

**Purpose.** Provides the minimal steps to recreate the datasets, train the model, regenerate figures, and run the UI.

## Appendix F.1 — Reproducibility

* accident\_full\_weather\_detailed.csv — Accident rows joined to hourly weather.
* non\_accident\_weather\_balanced11111.csv — Matched locations, different times.
* all\_data\_.csv — Unified modelling table (balanced classes, engineered features).

If you use different filenames, update the paths inside main\_data\_preparation.py.

## Appendix F.2 — Environment

python -V # tested on Python 3.11

pip install -r requirements.txt

# key packages:

# pandas, numpy, scikit-learn, xgboost, lightgbm, catboost, joblib, django, requests

## Appendix F.3 — Recreate features & dataset

python code/main\_data\_preparation.py \

--acc data/accident\_full\_weather\_detailed.csv \

--nonacc data/non\_accident\_weather\_balanced11111.csv \

--out data/all\_data\_.csv \

--seed 42

Ensures **28-feature** schema and constant fill **−1** for unknowns.

Saves train/test split indices (seed **42**) for repeatability.

## Appendix F.4 — Train models & export the deployed estimator

python code/main\_model\_training.py \

--data data/all\_data\_.csv \

--out models/accident\_model.pkl \

--model xgb \

--seed 42

Trains Decision Tree, Random Forest, LightGBM, CatBoost, **XGBoost**; logs **AUC/ACC/P/R/F1**; writes plots to /outputs/.

Exports the **XGBoost** model configured in **Table D.1** to models/accident\_model.pkl.

## Appendix F.5 — Regenerate figures(Chapter 5 Appendix B)

**ROC & PR curves:** saved by main\_model\_training.py to /outputs/roc.png, /outputs/pr.png.

**Confusion matrices:** /outputs/cm\_<model>.png (DT/RF/LGBM/CatBoost); XGBoost CM is in Chapter 5.

**PCA diagnostics & generalisation gap:** terminal screenshots saved as images (see Appendix B).

## Appendix F.6 — Run the UI(Django)

cd ui/accident

python manage.py migrate

python manage.py runserver 8000

Place models/accident\_model.pkl under /ui/accident/models/ (or adjust core/model.py).

The form builds the **28 features** in the fixed order and shows **class** and **probability** with a risk band.

## Appendix F.7 — Determinism & seeds

All scripts use random\_state=42 where relevant; results may vary slightly with library versions or thread settings.

For stricter determinism, set environment variable OMP\_NUM\_THREADS=1 and n\_jobs=1 in training.

## Appendix F.8 — Hardware used (informative)

Trained on commodity CPU laptop; no GPU required. XGBoost with tree\_method="hist" and n\_estimators=5000 fits in memory with the balanced dataset.

## Appendix F.9 — Hardware used (informative)

Replace STATS19 + vehicle files for the target region/year; re-run the weather join (same API).

Rebuild negatives with the same **location-matched, time-shifted** approach.

Run F.4 → F.5; verify metrics and export the new accident\_model.pkl.

Update the UI **feature list** only if the serving schema changes (not recommended).

# REFERENCES

Alotaibi, A.S. (2018) ‘Density-based clustering for road accident data analysis’, *International Journal of Advanced and Applied Sciences*, 5(8), pp. 113–121.

Anselin, L. (1995) ‘Local Indicators of Spatial Association—LISA’, *Geographical Analysis*, 27(2), pp. 93–115.

Becker, N., Rust, H.W. and Ulbrich, U. (2022) ‘Weather impacts on various types of road crashes: a quantitative analysis using generalized additive models’, *European Transport Research Review*, 14(1), p. 37.

Blows, S., Ivers, R.Q., Woodward, M., Connor, J., Ameratunga, S. and Norton, R. (2003) ‘Vehicle year and the risk of car crash injury’, *Injury Prevention*, 9(4), pp. 353–356.

Breiman, L. (2001) ‘Random forests’, *Machine Learning*, 45(1), pp. 5–32.

Chen, T. and Guestrin, C. (2016) ‘XGBoost: A scalable tree boosting system’, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

Department for Transport (YEAR) *STATS19 road safety data* [Dataset]. Department for Transport. Available at: URL (Accessed: Day Month YEAR).

Dorogush, A.V., Ershov, V. and Gulin, A. (2018) ‘CatBoost: gradient boosting with categorical features support’, *arXiv preprint* arXiv:1810.11363.

Iqbal, A., Rehman, Z.U., Ali, S., Ullah, K. and Ghani, U. (2020) ‘Road Traffic Accident Analysis and Identification of Black Spot Locations on Highway’, *Civil Engineering Journal*, 6(12), pp. 2448–2456. doi:10.28991/cej-2020-03091629.

Jones, A. and Brown, B. (2017) ‘Weather and Road Accident Severity: A Logistic Regression Approach’, *Transportation Research Part F: Traffic Psychology and Behaviour*, 45, pp. xx–yy.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.Y. (2017) ‘LightGBM: A highly efficient gradient boosting decision tree’, in *Advances in Neural Information Processing Systems*, 30.

Kumar, R., Singh, A. and Mehta, P. (2017) ‘Driver History and Accident Risk Analysis in Urban India’, *Journal of Transport Safety and Security*, xx(x), pp. xx–yy.

Lobo, A., Ferreira, S., Iglesias, I. and Couto, A. (2019) ‘Urban road crashes and weather conditions: untangling the effects’, *Sustainability*, 11(11), p. 3176.

McCartt, A.T., Mayhew, D.R., Braitman, K.A., Ferguson, S.A. and Simpson, H.M. (2009) ‘Effects of age and experience on young driver crashes: review of recent literature’, *Traffic Injury Prevention*, 10(3), pp. 209–219.

Open-Meteo (YEAR) *API documentation*. Open-Meteo. Available at: URL: https://www.data.gov.uk/dataset/cb7ae6f0-4be6-4935-9277-47e5ce24a11f/road-accidents-safety-data (Accessed: Day Month YEAR).

Perrels, A., Votsis, A., Nurmi, V. and Pilli-Sihvola, K. (2015) ‘Weather conditions, weather information and car crashes’, *ISPRS International Journal of Geo-Information*, 4(4), pp. 2681–2703.

Silverman, B.W. (1986) *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall.

Smith, J. and Taylor, M. (2018) ‘Weather effects on traffic accidents: a time-series analysis’, *Journal of Transportation and Safety*, 33(4), pp. 245–259.

Szenasi, S., Sipos, M. and Mogyorosi, P. (2022) ‘PSO-based optimization of DBSCAN algorithm parameters for road accident blackspot localization’, in *2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC)*, Reykjavík, Iceland, pp. 000043–000048. doi:10.1109/ICCC202255925.2022.9922768.

Wang, D., Yulang, H. and Cai, Z. (2023) ‘A two-phase clustering approach for traffic accident black spots identification: integrated GIS-based processing and HDBSCAN model’, *International Journal of Injury Control and Safety Promotion*, 30, pp. 1–12. doi:10.1080/17457300.2022.2164309.

Zhao, Y., Lin, C. and He, Z. (2022) ‘Ensemble learning models for road traffic accident prediction’, *Expert Systems with Applications*, 199, p. 116947.