TECHNISCHE UNIVERSITÄT WIEN

Data management plan (DMP)

Predicting Road Accident Severity in Great Britain

uk-collision-severity

Version	Effective date	Description of document/changes
1.0	09/04/2026	First version of the DMP – created for the start of the project
2.0	30/05/2026	Final version of the DMP

Level of distribution		

FWF Data Management Plan (DMP)

<table>
<tr><th colspan="2">I General Information</th></tr>
<tr><td>I.1 Administrative information</td><td>
PI:Yehea El Dib, e12450748@student.tuwien.ac.at, TU Wien, ROR: ror.org/04d836q62, Principal Investigator

DMP version: 2.0, 30/05/2026

Contributors:

Charles Logan, e12550259@student.tuwien.ac.at, TU Wien, ROR: ror.org/04d836q62, Project Member

Julian Hardt, e12330562@student.tuwien.ac.at, TU Wien, ROR: ror.org/04d836q62, Project Member

Balthasar Höfinger, e11908607@student.tuwien.ac.at, TU Wien, ROR: ror.org/04d836q62, Project Member
</td></tr>
<tr><td>I.2 Data management responsibilities and resources</td><td>
Person responsible for data management and DMP: Yehea El Dib, e12450748@student.tuwien.ac.at, TU Wien, ROR: ror.org/04d836q62

Co-ordination of data management responsibilities across partners: Yehea El Dib, e12450748@student.tuwien.ac.at, TU Wien, ROR: ror.org/04d836q62

Resources costed in for RDM: There are no costs dedicated to data management and ensuring that data will be FAIR.
</td></tr>
<tr><th colspan="2">II Data Characteristics</th></tr>
<tr><td>II.1 Data description and collection or re-use of existing data</td><td>
Produced datasets:
<table>
<tr><th>dataset ID</th><th>title</th><th>type</th><th>format</th><th>estimated volume</th><th>contains sensitive data</th><th>description</th></tr>
<tr><td>P1</td><td>Input data</td><td>Structured text</td><td>CSV</td><td>100 - 1000 MB</td><td>no</td><td>Police-reported road traffic collision records from Great Britain (2020–2024),
collected via the STATS19 reporting system and published by the UK Department
for Transport. ~503,000 rows with 44 attributes per collision.

The raw CSV was downloaded once from the UK Government open data portal and</td></tr>
</table>
</td></tr>
</table>

						then imported into a 3NF relational schema in TU Wien's DBRepo. Source URL: https://www.gov.uk/government/statistical-data-sets/road-safety-open-data DBRepo: https://test.dbrepo.tuwien.ac.at/database/82c19b39-246c-4409-b25c-8baf3a158a70 DBRepo DOI: 10.82556/c8r3-bf26 Licence: Open Government Licence v3.0
	P2	Processed data splits (train/validation/test)	Structured text	CSV	100 - 1000 MB	no	Cleaned and split versions of the input data, produced by 01_load_data.py and 02_preprocess.py: - train.csv — 70% of the cleaned rows, stratified by collision_severity - train_resampled.csv — same training set after SMOTE oversampling of the minority classes - validation.csv — 15%, used for model selection (Macro F1) - test.csv — 15%, held out and only used at final evaluation All splits use a fixed random_state=42 for reproducibility. Each file contains the 15 ML features plus the collision_severity label (Fatal / Serious / Slight) — 16 columns total. Licence: CC BY 4.0
	P3	Trained Gradient Boosting collision severity model	Other	joblib (.pkl)	100 - 1000 MB	no	Serialised scikit-learn GradientBoostingClassifier trained on the SMOTE-resampled training set to predict collision severity (Fatal / Serious / Slight). Selected from three candidates (Decision Tree, Random Forest, Gradient Boosting) by Macro F1 score on the validation set.

						Hyperparameters: n_estimators=150, max_depth=6, learning_rate=0.1, random_state=42. Test set performance: accuracy 0.676, macro F1 0.403. Per-class F1: Fatal 0.117, Serious 0.288, Slight 0.805. Detailed metadata is in docs/fair4ml/gradient_boosting_severity_2026-05-25.json and docs/model-card.md. Filename: gradient_boosting_severity_2026-05-25.pkl Licence: CC BY 4.0
P4	Model evaluation outputs (predictions and figures)	Structured text, Images	CSV, PNG	100 - 1000 MB	no	Outputs produced by 03_train_classifier.py and 04_evaluate.py when the model is evaluated on the held-out test set: - test_predictions_2026-05-25.csv — predicted vs. actual labels for all 17,220 test samples - 01_data_understanding.png — feature distributions and class balance overview - 02_class_imbalance.png — class distribution before vs. after SMOTE - 03_confusion_matrix.png — confusion matrix on the test set - 04_performance_comparison.png — per-class precision/recall/F1 + the three-model comparison on the validation set - 05_feature_importance.png — feature importances from the Gradient Boosting model Licence: CC BY 4.0

Reused datasets:

dataset ID	title	source	rights (e.g. license)	contains sensitive data	description
R1	STATS19 Road Safety Open Data	https://www.gov.uk/government/statistical-data-sets/road-safety-open-data		no	Police-reported road traffic collision records for Great Britain (2020–2024), collected via the STATS19 reporting system and published by the UK Department for Transport. Contains ~503,000 rows with 44 attributes per collision including location, severity, road conditions, and weather. Downloaded once from the UK Government open data portal and imported into TU Wien DBRepo.

Methods and software used for data generation and reuse

The STATS19 dataset was re-used as the input to the ML pipeline. We downloaded the raw CSV once from the UK Government open data portal (OGL v3.0) and imported into TU Wien's DBRepo. The data was processed with Python 3.13.11. Pandas was used for data manipulation, scikit-learn 1.8.0 for the three classifiers (Decision Tree, Random Forest, Gradient Boosting), imbalanced-learn for SMOTE on the training set, matplotlib for figures, and joblib for serialising the trained model. All dependencies are pinned in requirements.txt. Generated outputs — the train/validation/test splits, the trained model, the test-set predictions, and the evaluation figures — were produced by running the four pipeline scripts (01_load_data.py through 04_evaluate.py) in sequence.

III Documentation and Data Quality

III.1 Metadata and documentation

Data organisation, metadata, and documentation:

Code and metadata are version-controlled in Git on GitHub at https://github.com/b4lz2/uk-collision-severity-prediction. Every important change to the pipeline, metadata, or documentation is committed with a message that references the task it belongs to (e.g. "T2.3: unit mapping ..."). The final release was tagged on GitHub, which automatically minted a Zenodo DOI. Folder structure in the repository: - src/ — the four pipeline scripts (01_load_data.py through 04_evaluate.py) - src/data/processed/ — the train/validation/test splits - src/outputs/models/ — the trained model (.pkl) - src/outputs/predictions/ — test set predictions - src/outputs/figures/ — evaluation plots - notebooks/ — Jupyter notebooks for DBRepo setup (unit mapping, data load) - docs/ — metadata files,

	model card, standards analysis The input data lives in TU Wien's DBRepo in a 3NF schema, with the collisions fact table and 17 lookup tables. Every column has a stable UUID inside DBRepo, so column references don't break if names change. Files follow ISO 8601 date conventions (gradient_boosting_severity_2026-05-25.pkl) and snake_case throughout. We provide five parallel metadata standards, covering different views: - RO-Crate (ro-crate-metadata.json): package-level metadata listing all files in the research object. - CodeMeta (codemeta.json): software metadata — authors, license, version, pinned dependencies, programming language. - FAIR4ML (docs/fair4ml/gradient_boosting_severity_2026-05-25.json): the trained model — algorithm, hyperparameters, evaluation metrics, training data reference, intended use, limitations. - Croissant (croissant.json): the dataset schema — every field with its data type and unit of measurement (SI Digital Framework URIs for metre and degree, QUDT URIs for mph, year, count, fraction). - Model Card (docs/model-card.md): the human-readable model documentation with use cases, out-of-scope use, ethical considerations. In addition, every numeric column in DBRepo has a unit_uri annotation (from T2.3) and a concept_uri annotation (from T2.2), both written via the DBRepo REST API. An analysis of how these standards overlap and where they conflict is in docs/standards-overlap-analysis.md. This will help others to identify, discover and reuse our data. Additionally, we will provide common metadata such as title, description, or keywords when publishing data in open access repositories. In such a case, we will follow the default template provided by the repository, such as Data Cite Metadata or Dublin Core. As far as possible, we will use controlled vocabularies for our data to allow inter-disciplinary interoperability and machine-actionability. The main reference is README.md in the repository root. It covers the project abstract, requirements (Python 3.13.11), installation, step-by-step reproduction (four scripts to run in order), inputs and outputs with paths and formats, the ER diagram of the DBRepo schema, the two SQL views used, DBRepo API endpoints with example URLs, authentication via environment variables, error handling, contributor list with ORCIDs, and the three licences. Each Python script (except 01_load_data.py) has a header comment block describing what it reads, what it writes, and the section structure of the script. The notebooks folder contains Jupyter notebooks documenting the DBRepo setup steps: schema creation, semantic mapping of categorical attributes, unit mapping of numeric attributes, view creation, and data loading. requirements.txt contains pinned versions of every direct dependency, so the experiment can be re-run in a fresh virtual environment by anyone. The trained model is documented twice: machine-readably in the FAIR4ML JSON, and human-readably in docs/model-card.md.
III.2 Data quality control	**Data quality control**: The following data quality checks will be done: standardised data capture, representation with controlled vocabularies and - A fixed random_state=42 is used everywhere (data splits, SMOTE, classifier initialisation), so anyone who reruns the pipeline gets exactly the same results. - Class imbalance in the training set is handled with SMOTE (training set only), so the model can learn from the minority Fatal class. - Row counts and class distributions are printed at every pipeline step to catch silent data loss. - The data loader verifies the row count of the DBRepo view against the X-Count response header before processing..
IV Data Storage, Sharing, and Long-Term Preservation	
IV.1 Data storage and backup during the research process	**Storage and backup facilities:** For the duration of the project, storage and backup of data will be ensured by Yehea El Dib (acting as the person responsible for data management and DMP) in cooperation with the system operator. The input data and database schema are stored on TU Wien's DBRepo infrastructure. Trained models and evaluation outputs are deposited in the TU Wien Research Data Repository (TUWRD). Code, metadata, and small data artefacts are version-controlled on

GitHub.

P1 (Input data), P2 (Processed data splits (train/validation/test)), P3 (Trained Gradient Boosting collision severity model), P4 (Model evaluation outputs (predictions and figures)) will be stored on GitHub. External storage will be used because the course explicitly requires a github repository for code, metadata, and small data artifacts, with public access for review and grading. the data itself is not sensitive (open government data under ogl v3.0), so there's no compliance reason to keep it on institutional storage. long-term preservation is handled separately via the tu wien research data repository, which is also linked here as a repository for publication.

Data security and protection of sensitive data:

We pay strict attention to compliance with the relevant institutional and national data protection policies. At this stage, it is not foreseen to process any sensitive data in the project. If this changes, advice will be sought from the data protection specialist at TU Wien, and the DMP will be updated.

Access to data during research:

dataset ID	selected project members	all other project members	the public
P1	reading only	reading only	reading only
P2	writing	reading only	reading only
P3	writing	reading only	reading only
P4	writing	reading only	reading only
R1	reading only	reading only	reading only

IV.2 Data sharing and long-term preservation

Data publication and access conditions:

As far as possible, obtained datasets will be published in repositories. Details on access conditions, reuse licenses, reasons for restrictions, etc. are collected in the table below.

dataset ID	access conditions	estimated publication date	location for publication (repository)	PID	license
P1	Open	2026-05-30	TU Wien Research Data	DOI	OGL-v3.0
P2	Open	2026-05-30	GitHub, TU Wien Research Data, Zenodo	DOI	CC-BY-4.0
P3	Open	2026-05-30	GitHub, TU Wien Research Data, Zenodo	DOI	CC-BY-4.0
P4	Open	2026-05-30	GitHub, TU Wien Research Data, Zenodo	DOI	CC-BY-4.0

Repository description:

GitHub is the best place to share code with friends, co-workers, classmates, and complete strangers. Over three million people use GitHub to build amazing things together. With the collaborative features of GitHub.com, our desktop and mobile apps, and GitHub Enterprise, it has never been easier for individuals and teams to write better code, faster. Originally founded by Tom Preston-Werner, Chris Wanstrath, and PJ Hyett to simplify sharing code, GitHub has grown into the largest code host in the world. https://github.com

ZENODO builds and operates a simple and innovative service that enables researchers, scientists, EU projects and institutions to share and showcase multidisciplinary research results (data and publications) that are not part of the existing institutional or subject-based repositories of the research communities. ZENODO enables researchers, scientists, EU projects and institutions to: easily share the long tail of small research results in a wide variety of formats including text, spreadsheets, audio, video, and images across all fields of science. display their research results and get credited by making the research results citable and integrate them into existing reporting lines to funding agencies like the European Commission. easily access and reuse shared research results. https://zenodo.org/

TU Wien Research Data is an institutional repository of TU Wien to enable storing, sharing and publishing of digital objects, in particular research data. It facilitates the funders' requirements for open access to research data and the FAIR principles by making research output findable, accessible, interoperable, and reusable. A DOI is assigned to each dataset published in TU Wien Research Data. This service is developed by the TU Wien Center for Research Data Management and hosted by TU.it. https://researchdata.tuwien.ac.at/

Methods or software needed to access and use data: Everything works with free, open-source software: - For the CSV outputs (splits, predictions): any tool that reads CSV — Excel, LibreOffice, pandas, R. - For the PNG figures: any image viewer. - For the trained model (.pkl): Python with scikit-learn 1.8.0 and joblib. Other versions of scikit-learn may load the file but are not guaranteed to behave identically. - For re-running the full pipeline: Python 3.13.11 with

the dependencies pinned in requirements.txt, plus a DBRepo account to fetch the input data via the REST API. - For the metadata files: any JSON-LD viewer or a plain text editor. No proprietary software is needed at any step.

Long-term preservation and deletion of data:

dataset ID	location for long-term storage	minimum retention period (≥ 10 years)	foreseeable research uses and/or users
P1	TU Wien Research Data	10 years	This is a course project, so the primary audience is the DaSt course team (lecturers, reviewers) and other students in the same course. Beyond that, the data and outputs could be useful to: - Other students learning about FAIR data practices, especially the combination of multiple metadata standards (RO-Crate, CodeMeta, FAIR4ML, Croissant, Model Card) on the same project. - Researchers working on UK road safety who want a reproducible baseline on the STATS19 data using a normalised 3NF schema in DBRepo. - People interested in how to wire an ML pipeline directly to a DBRepo REST API instead of reading local CSVs. The trained model itself is not useful for any real-world risk scoring or policy decisions — the limitations are documented in the model card and FAIR4ML metadata.
P2	GitHub, TU Wien Research Data, Zenodo	10 years	
P3	GitHub, TU Wien Research Data, Zenodo	10 years	
P4	GitHub, TU Wien Research Data, Zenodo	10 years	

V Legal and Ethical Aspects

V.1 Legal aspects

Personal data:

At this stage, it is not foreseen to process any personal data in the project. If this changes, advice will be sought from the data protection specialist at TU Wien, and the DMP will be updated.

Intellectual property rights and rights of use:

The following individual(s) hold rights and control access to the project data: All four project members have writing access to datasets via the TU Wien Research Data Repository and DBRepo. The trained model and evaluation outputs are owned and managed by the group. The input data (STATS19) is owned by the UK Department for Transport — we only re-use it under the Open Government Licence v3.0

we do not control access to the original source.

V.2 Ethical aspects	**Ethical issues:** No particular ethical issue is foreseen with the data to be used or produced by the project. This section will be updated if issues arise.