Module Title: Level 8 H. Dip in Data Analytics for Business

Assessment Title: Machine Learning

Lecturer: Dr Muhammad Iqbal

Student Name: Xiu(Janet) Shi

Student Number: sba23167

Submission Date: 15th April 2025

**UK Road Accident Severity Prediction: Machine Learning Analysis and Classification**

Executive Summary

Road traffic accidents represent a significant public health concern globally, with substantial economic and social costs. In the UK, understanding and predicting the severity of road accidents is essential for developing effective safety measures and policies. This report analyzes the 2010 UK road accident dataset to identify patterns and develop predictive models for accident severity.

The analysis employs machine learning techniques to classify accidents into binary categories: "slight" severity versus "serious/fatal" severity. This classification framework enables the application of established machine learning approaches to road safety, where preventing progression from slight to serious/fatal outcomes is the primary goal.

Due to computational constraints, this project utilized random sampling to reduce data size for model experimentation and validation. Multiple sample sizes were tested, ranging from 5,000 to 30,000 rows. This approach enabled faster iteration while maintaining representation of the class distributions and preserving the integrity of the dataset.

Data source:

<https://www.kaggle.com/datasets/tsiaras/uk-road-safety-accidents-and-vehicles>

**Accident\_Information.csv:** every line in the file represents a unique traffic accident (identified by the Accident\_Index column), featuring various properties related to the accident as columns. Date range: 2005-2017

Due to the volume of the Accident\_Information.csv and local computing power. Dataset has been intentionally filtered to include data of in Year 2010 only (~115k rows, 34 features).

**Acknowledgements**

Thanks to data.gov.uk for making this information publicly available. Also thanks to Dave Fisher-Hickey for previously publishing, what I consider to be, the first solid and structured version of this dataset on Kaggle.

**Data Understanding and Preprocessing**

**Dataset Characteristics**

The UK road accident dataset contains numerous features related to accident circumstances, including temporal factors (time, day, month), environmental conditions (weather, light, road surface), location characteristics (road type, urban/rural), and accident details (number of vehicles, casualties). The target variable is accident severity, which has been grouped into a binary classification problem: slight (0) versus serious/fatal (1).

A key characteristic of this dataset is the class imbalance, with slight accidents significantly outnumbering serious/fatal ones. This imbalance necessitates careful consideration during model selection and evaluation to avoid bias toward the majority class.

**Feature Analysis**

Before dive into Data Processing and Model development, I conducted a thorough exploratory data analysis to understand the UK road accident dataset characteristics and identify patterns that might influence accident severity. This analysis guided my feature selection and preprocessing decision, evaluate the choice of different models of the machine learning approaches.

**Class Distribution Analysis**

The initial exploration was to identify different severity of accident distribution.

A graph of a number of objects

AI-generated content may be incorrect.

The chart clearly shows that “Slight” accidents outnumber “Serious” and “Fatal” accidents. This imbalance is typical in road safety data but presents a challenge for predictive modelling. Based on this characteristics I decided to group “Serious” and “Fatal” accidents together to create a binary classification problem.

## Create binary target: 0 = Slight (loyal), 1 = Serious or Fatal (churn)

df["Severity\_Binary"] = df["Accident\_Severity"].map({"Slight": 0, "Serious": 1, "Fatal": 1})

**Other distributions and patterns**

I then analyzed how accidents distribute across different time to identify potential patterns:

A graph of a number of accidents

AI-generated content may be incorrect.

A graph of a number of injuries

AI-generated content may be incorrect.

**A graph of a speed limit

AI-generated content may be incorrect.**

**A graph of a accident

AI-generated content may be incorrect.**

**A graph of a graph with text

AI-generated content may be incorrect.**

**Accident Geographic Context**

The urban and rural comparison shows that rural areas have a higher proportion of serious and fatal accidents despite having fewer accidents overall.

**A graph of a accident

AI-generated content may be incorrect.**

**Feature Importance Analysis**

**A graph of a number of blue bars

AI-generated content may be incorrect.**

To assess feature relevance, after applying PCA for dimensionality reduction, the feature importance chart shows that the first few principal components capture most of the predictive information. It confirmed that PCA was effectively preserving the most important variance in the data while reducing dimensionality.

**Feature Selection Decisions**

Based on above exploratory analysis, I made decisions for the feature selection and preprocessing:

* The number of vehicles, road types, speed limit, road conditions and light conditions showed strong relationships with severity
* The hour and month showed patterns related to accident severity
* The Urban/Rual geographic context proved importance to predict accident severity

A screen shot of a graph

AI-generated content may be incorrect.

**Train-Test Data Split Justification**

The entire dataset contains ~115k rows, we have intentionally reduced to a smaller size ‘max\_iter=1000’ to evaluate models’ performance and generalizability. Different train-test split percentage is explored. 5%, 10%, 20% of data were held out for testing while the remaining data was used for training. Surprisingly all these split percentage produced very similar results in terms of accuracy. All approximately 85.9%. 5% split yielding a slightly higher score!!

Despite the close performance, the 80/20 split was ultimately selected for the final model evaluation for a number of reasons: The industry convention ( 80/20 split ). Generalizability: a larger test set (20%) provides a more comprehensive view of model performance. Avoiding overfitting to small test sets, such 5% might lead to inflated accuracy due to lower data diversity in the test set. Also sufficient training data at 80% ensures the model has enough samples to learn meaningful patterns, particularly valuable in imbalanced classification problems like accident severity.

**Preprocessing Pipeline**

The preprocessing pipeline implemented in the code addresses several critical aspects:

* **Missing Value Handling**: Different strategies are applied based on feature type:
* **Feature Encoding**: Categorical features are one-hot encoded to convert them into a format suitable for machine learning algorithms.

## Update preprocessing pipeline, we prepare one-hot encoding to convert categorical variables to numerical input

# Output must be converted to dense arrays to be compatible for PCA

## Before implementing PCA, we have already applied one-hot encoding to convert categorical variables to numerical input

## Apply PCA to reduce dimensionality, the goal is to reduce noisy features retaining most of the data variance

## Retained 95% of variance

## Before implementing PCA, we have already applied one-hot encoding to convert categorical variables to numerical input

categorical\_pipeline = Pipeline([

("imputer", SimpleImputer(strategy="most\_frequent")),

("encoder", OneHotEncoder(handle\_unknown="ignore", sparse\_output=False))

])

numerical\_pipeline = Pipeline([

("imputer", SimpleImputer(strategy="median")),

("scaler", StandardScaler())

])

## Apply PCA to reduce dimensionality, the goal is to reduce noisy features retaining most of the data variance, retained 95% of variance

from sklearn.decomposition import PCA

pca = PCA(n\_components=0.95, random\_state=42)

X\_train\_pca = pca.fit\_transform(X\_train\_proc)

X\_test\_pca = pca.transform(X\_test\_proc)

## The new dataset X\_train\_pca will have fewer columns but still carry most of the important features

* **Feature Scaling**: Numerical features are standardized to ensure that features with larger scales don't dominate the model training process. This is particularly important for distance-based algorithms like SVM.

The preprocessing pipeline approach is to ensure data quality and preparing it for effective model training. The choice of preprocessing steps directly influences model performance and is tailored to the characteristics of the dataset.

**Model Selection Justification**

The selection of appropriate machine learning models for accident severity prediction is guided by the nature of the binary classification, data characteristics. This section provides a logical justification for the models selected, also consider the predictive performance with interpretability and computational feasibility.

Given the binary target variable “Slight” vs “Serious+Fatal” accidents. The classification models were selected to determine the probability of a case belonging to the “Churn” (Serious+Fatal) class. Based on these criteria, three primary models were selected for comparative evaluation: Logistic Regression Support Vector Machine(SVM) and Random Forest. Each was chosen to highlight different strengths relevant to the dataset and prediction challenge.

I have also implemented a timer in each of the blocks to track the run time to evaluate the model performance as well as accuracy.

1. **Logistic Regression (LR)**

In our use-case, LR serves as a baseline classifier. It’s simple and widely used in binary classification. Evaluation on a random 5,000 row of data sample showed reasonable predictive justification.

lr = LogisticRegression(max\_iter=5000, random\_state=42)

1. **Support Vector Machine (SVM)**

SVMs are effective for non-linear classification. In our use-case, both linear and RBF(non-linear) kernels were tested. The SVM model used a balanced class with to mitigate class imbalance.

svm = SVC(kernel="linear", probability=False, max\_iter=5000, random\_state=42)

*## Define and train SVM model (for classification of churn vs loyal) the non-linear kernels*

svm\_model = SVC(kernel="rbf", # hyperparameter: kernel type

probability=True, # enables probability estimates

C=1.0, gamma="scale", # hyperparameter: regularization strength

class\_weight="balanced", # hyperparameter: kernel coefficient

random\_state=42) # handles class imbalance

1. **Random Forest (RF)**

Random Forest is an ensemble-based tree model that combines multiple decision trees to improve robustness and reduce overfitting. It can handle both categorical and numerical data, and provides feature importance metrics.

rf\_model = RandomForestClassifier(

n\_estimators=100,

class\_weight="balanced

max\_features="sqrt

random\_state=42

n\_jobs=-1

)

**Purpose of Hyperparameter Tuning in Machine Learning**

Hyperparameter tuning (*model hyperparameters*) is the process of finding the best combination of model parameters that are not learned from the data but control how the model learns from the data. I tried to think hyperparameters are like those dials and knobs in a music studio’s central control panel. In our case, each dial and knob represent a particular parameter in the model.

The primary purpose of hyperparameter tuning is to optimize models’ performance by carefully adjusting different parameters. The goal is to optimize model performance by carefully adjusting those dials and knobs to achieve the best audio performance in the studio.

The results are often reflecting in the model’s accuracy, generalization of unseen data, reduce overfitting or underfitting, improve overall F1, ROC-AUC or relevant scores.

**Model Learning Behaviour: Underfitting Overfitting and Generalization**

In supervised machine learning, especially in classification task like our use-case. Accident severity (Slight vs. Serious/Fatal), the ultimate goal is not just to perform well on training data, it’s also important to make reliable predictions on unseen, real-world data. Such as same kind of data captured in different years. This is call generalization.

**Underfitting** occurs when a model is too simple or only limited data is captured without the full picture due to various of constrains. Using a model with insufficient diversity or limited training such as low ‘max\_item=100’ or missing key features etc. In our analysis, the Logistic Regression model with ‘max\_iter=5000’ give the volume of our dataset

LogisticRegression(max\_iter=5000, random\_state=42)

However if we only set ‘max\_iter=100’, this would fail to converge leading to poor training accuracy at 60% and poor test accuracy at ~58% - which shows us that the model couldn’t find decision boundaries to seprate “Slight” from “Serious/Fatal”

LogisticRegression(max\_iter=100, random\_state=42)

**Overfitting** is the opposite, it occurs when a model learns the noise or random fluctuations in the training data. When there are too many features, random data points without regularization or data boundraies, it could result in high training accuracy but low test accuracy. In our analysis, the Random Forest with no restriction

(max\_depth=None, large n\_estimators)

would lead to high training accuracy at 98% and test accuracy at 66%. This shows the model over specialized on training cased that don’t reflect real-world senarios.

**Generalization:** not too high or low, just right. I see generalization as a good balance between complexity, flexibility and the right amount of characteristics that will work well across the entire datasets.

**3 Models Performance Comparison**

The 3 models **Logistic Regression**, **Support Vector Machine (SVM)**, and **Random Forest** are put to the cross validation. To ensure fair evaluation, a 5 fold cross validation was applied on a representative sample of 30,000 records, which helped reduce training time while preserving class balance. Before training, all features were preprocess using one-hot encoding and standard scaling, followed by Principle Component Analysis (PCA) to reduce dimentionality while retaining 95% of the data variance.

A screen shot of a computer

AI-generated content may be incorrect.

The Logistic Regression showed the highest accuracy, but low ROC-AUC and almost 0 F1score. SVM (RBF Kernel) performed similarly in ROC-AUC and F1 score but might need more computation and tuning due to the non-linear nature. Random Forest showed decent accuracy but lagged behind in F1score and ROC-AUC. This may due to the tree-based many splits and potentially overfitting the dataset even after PCA was applied.

A graph with different colored bars

AI-generated content may be incorrect.

**Conclusion**

In my opinion, the Logistic Regression may only be a good starting point due to the high accuracy score. SVM however has the most balanced trade-off between F1 and ROC-AUC scores. In my understanding, SVM model have better balance performance across key metrics F1 Score and ROC-AUC. This is valuable in our use-case where predicting the minority class of Serious or Fatal accidents is critical. SVM model stand out as a robust model for further consideration and future tuning.

**Final Thoughts and Academic Backing**

Throughout this project, I’ve explored different models, evaluated them with real data, and gained deeper insights into how each approach performs in predicting accident severity. What’s more reassuring is that two recent academic studies align strongly with the choices I made.

One study compared traditional logistic regression with hybrid machine learning models like SVM combined with optimization algorithms. (Haghshenas et al., 2025). Another large-scale UK study also compared Random Forest, Logistic Regression, Naïve Bayes, and ANN, showing that Random Forest and Logistic Regression achieved the highest prediction accuracy (Obasi and Benson, 2023).

This alignment between practical results and academic literature gives me confidence that the approaches used in this analysis are well-grounded. It also reflects the depth of research and experimentation that went into ensure that the conclusion drawn here are credible, not just computational.

**References**

Kaggle, 2024. UK Road Safety: Traffic Accidents and Vehicles. [online] Available at: <https://www.kaggle.com/datasets/tsiaras/uk-road-safety-accidents-and-vehicles> [Accessed 10 Apr. 2025].

Scikit-learn, 2024. sklearn.decomposition.SparsePCA — scikit-learn 1.4.1 documentation. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.SparsePCA.html> [Accessed 10 Apr. 2025].

Scikit-learn, 2024. *sklearn.decomposition.SparsePCA — scikit-learn 1.4.1 documentation*. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.SparsePCA.html> [Accessed 15 Apr. 2025].

Amazon Web Services (n.d.) *What is Hyperparameter Tuning?* [online]. Available at: <https://aws.amazon.com/what-is/hyperparameter-tuning/> [Accessed 15 Apr. 2025].

IBM (n.d.) Overfitting vs. underfitting: Finding the balance. [online] Available at: <https://www.ibm.com/think/topics/overfitting-vs-underfitting> [Accessed 15 Apr. 2025].

**ScienceDirect article (Transportation Engineering):**  
Haghshenas, S.S., Guido, G., Haghshenas, S.S. and Astarita, V., 2025. Predicting the level of road crash severity: A comparative analysis of logit model and machine learning models. *Transportation Engineering*, *20*, p.100323. Available at: <https://doi.org/10.1016/j.treng.2025.100323> [Accessed 15 Apr. 2025].

**Heliyon article (2023):**  
Obasi, I.C. and Benson, C., 2023. Evaluating the effectiveness of machine learning techniques in forecasting the severity of traffic accidents. Heliyon, 9(8), p.e18812. Available at: <https://doi.org/10.1016/j.heliyon.2023.e18812> [Accessed 15 Apr. 2025].