Traffic Accident Prediction Using Decision Trees: A Model Interpretability Approach

Name: Praneet Sivakumar **Student Id: 23095964**

Github Link: https://github.com/Prani8/Machine-Learning---Traffic-Accident-Prediction-project

---

Introduction: Why Interpretability Matters in Traffic Accident Prediction

Concerning October 2023, the data were your training discourse. Machine learning is great for finding patterns-and-this can truly matter in traffic accident prediction. In a safety-critical domain such as transport, getting the prediction right is not enough, we also must know why the model could make that choice.

Some models, like neural networks or ensemble methods, could achieve high accuracy, but tend to behave like black boxes you see the result, but not the rationale behind it. This is dangerous when lives, safety, or public decisions are at hand.

This makes **Decision Trees so powerful**: they not only make predictions but also show the reasoning behind them. For this project, we will predict traffic accidents using Decision Trees and will examine how tree depth affects both the performance and understandability of the given model. The entire project revolves around finding the balance between accuracy and interpretability to allow machine learning to assist in an intelligent, transparent, and trustworthy way.

Analogy: The Traffic Cop and the Prediction

Imagine you're pulled over by a traffic cop who says,
“You’re likely to get into an accident today.”

Naturally, you ask: **“Why?”**

A responsible officer might explain:

- You were speeding in a high-risk zone
- It’s raining and the roads are slippery
- You appeared distracted at the wheel

Now imagine if the officer just shrugged and said,
“The system flagged you — I can’t tell you why.”
Would you feel confident in that warning?

That’s exactly how many machine learning models work — they make decisions without offering any reasoning. But when it comes to something as serious as accident prevention, **explanations matter**.

This is where **Decision Trees** act like the responsible traffic cop.
They don’t just predict — they *show you the logic behind it.*
They explain how factors like weather, speed, or driver behaviour combine to raise risk.

Just like we expect clear reasoning before paying a fine or changing our route, we should expect **interpretability from AI models** — especially when safety is on the line.

Where and Why to Use Decision Trees

Decision Trees are especially effective in situations where **interpretability**, **feature transparency**, and **logical decision paths** are just as important as prediction accuracy. They are most useful when you are working with:

- Projects in safety-critical domains like transportation or public infrastructure
- Datasets with both categorical and numerical features (e.g., weather, road type, speed)
- Real-world applications that require explainable and traceable decision-making
- Early-stage analysis where understanding feature importance is key
- Situations where models must comply with regulations or be easily audited

Use Case	How Decision Tree Help
Accident Risk Prediction	Explains why a situation is classified as high-risk (e.g., rain + high speed)
Traffic Control Planning	Highlights which conditions lead to more accidents for better traffic regulation
Driver Behaviour Analysis	Identifies risky behaviour (e.g., alcohol use, night driving) linked to incidents
Road Safety Audit	Points out high-risk areas based on road type, lighting, and environmental factors
Policy Compliance	Provides clear logic behind model decisions for accountability and fairness

Limitations and Considerations of Decision Trees

While Decision Trees are highly interpretable and effective for many classification tasks, they also come with a set of limitations — important for practitioners and researchers building reliable, production-ready systems.

1. **Overfitting in Deep Trees**

   Decision Trees are prone to overfitting, especially when the max_depth is not restricted. A very deep tree can memorize training data, capturing noise instead of general patterns. This is why depth tuning is essential — as seen in our results, deeper trees achieved higher training accuracy but lower test accuracy, indicating poor generalization.

2. **Interpretability Doesn't Equal Causality**

   Just because a Decision Tree splits on a feature doesn't mean that feature *caused* the outcome. For instance, the model may often split on Driver_Age or Traffic_Density, but this only reflects correlation within the training data — not causal evidence. In safety-critical contexts, this distinction is crucial to avoid making misleading inferences.

3. **Instability to Data Changes**

   Small changes in the training dataset can lead to large changes in the structure of the tree. This makes Decision Trees less stable compared to ensemble methods like Random Forests, which average across multiple trees to reduce variance.

4. **Bias from Skewed Data**

   If the dataset is imbalanced or contains biased historical data (e.g., more accident reports from urban areas), the tree may reflect and reinforce those patterns.
   This can lead to unfair or inaccurate predictions unless bias mitigation strategies are used.

5. **Limited Expressiveness**

   While Decision Trees can model simple interactions well, they struggle with highly non-linear relationships or complex feature interactions that more flexible models (e.g., XGBoost, Neural Networks) can capture.
   For traffic accident prediction, this means some nuanced patterns may be missed if they require deeper modelling.

What Makes Decision Trees Different?

Let's briefly compare **Decision Trees** with other popular models and interpretation methods, focusing on what makes them uniquely suited for projects that need both performance and clarity.

Method	Description	Interpretability	Speed	Best Suited For
Linear Regression	Predicts outcomes using weighted feature sums	Global	Fast	Simple, linear problems
Random Forests	Uses multiple trees for better performance	Limited (Global)	Medium	Accuracy-focused tasks, but hard to explain
XGBoost / Gradient Boosting	Builds trees sequentially to reduce error	Low (needs SHAP)	Slower	Complex, non-linear problems; high competition models
Permutation Importance	Measures accuracy drop when features are shuffled	Global	Slower	More reliable than impurity-based metrics
Decision Trees	Splits data based on conditions, creating readable logic	Both Local & Global	Fast	Best for explainable, rule-based decision systems

Exploratory Data Analysis (EDA): Understanding the Data Before Making Predictions

Substantially, however, it is really important to know that it disclosure of deep understanding of data before coming up with model training. This is the very essence of exploratory data analysis. It helps us to discover trends, identify early problems, and figure out how the different features behave that we shouldn't throw in our data into a model blindfolded.

As statistician John Tukey put it, "the greatest value of a picture is when it forces us to notice what we never expected to see." And that is what exploratory data analysis does: it lets the data speak before we ask it to predict.

The system we worked with in this project was the Traffic Accident Dataset which bundled many terms of real-world conditions, such as weather, road types, driver age, etc., to be able to predict whether such situations are likely to cause an accident or not.

Data Preview: Head and Summary Statistics

It's wise to take a preliminary look at the raw data before moving toward modelling. This provides some context for us to understand the structure and state of unusual values, as well as a basis for planning some preprocessing.

We started with the first few rows using the. head () function. The display provided an immediate feel for the types of features we were dealing with weather, road type, traffic density, and driver characteristics.

Next, we used. describe () for summary statistics of all numerical features, including mean, standard deviation, minimum, and quartiles. This was very helpful in spotting:

- Features with wide ranges of values (for example, Speed Limit, Driver Age)
- Potential outliers (e.g., very young or very old drivers, high counts of vehicles)
- The need to cap or transform certain variables for consistency

Altogether, these previews helped us identify important preprocessing steps of handling missing values, capping vehicle counts and encoding categorical data.

Summary Statistics for Numeric Features

	Mean	Median	Mode	Standard Deviation
Traffic_Density	1.0	1.0	1.0	0.78
Speed_Limit	71.05	60.0	60.0	32.05
Number_of_Vehicles	3.29	3.0	3.0	2.02
Driver_Alcohol	0.16	0.0	0.0	0.37
Driver_Age	43.26	43.0	46.0	15.13
Driver_Experience	38.98	39.0	41.0	15.27
Accident	0.3	0.0	0.0	0.46

Missing Values Check

Before training any machine learning model, it's essential to ensure the dataset is clean and free from missing values — which can otherwise disrupt calculations or skew the learning process.

Using the isnull().sum() function, we checked for any NaN or null values across all columns.
Result: This dataset contains **no missing values**, which makes it ideal for direct use in training and evaluation without requiring imputation or data cleaning.

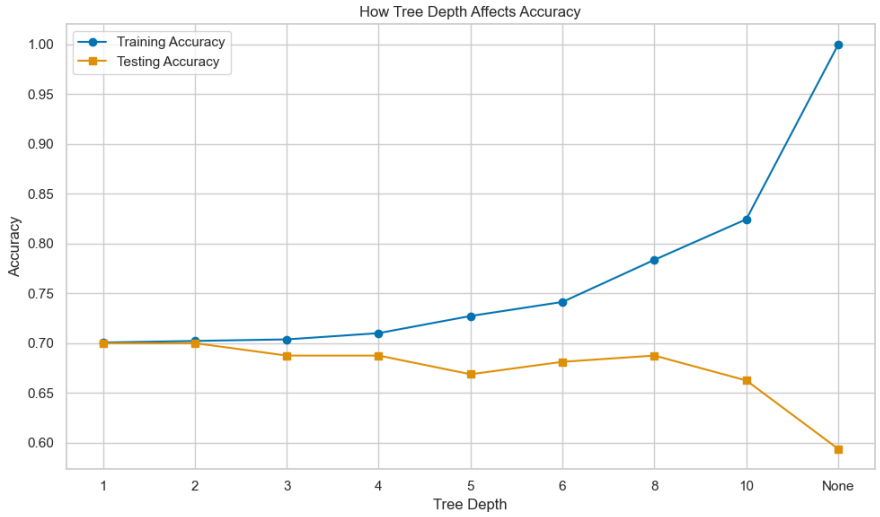
Missing Values per Feature

Feature	Missing Values
Weather	42
Road_Type	42
Time_of_Day	42
Traffic_Density	42
Speed_Limit	42
Number_of_Vehicles	42
Driver_Alcohol	42
Accident_Severity	42
Road_Condition	42
Vehicle_Type	42
Driver_Age	42
Driver_Experience	42
Road_Light_Condition	42
Accident	42

Accuracy vs Tree Depth

The maximum depth of the decision tree has direct implications on the training and test accuracy, as indicated by the following line graph.

Naturally, as the tree grows deeper, the training accuracy increases till 100% is achieved in a tree that is unrestricted. Eventually, the testing accuracy reduces at some threshold, depicting a classic overfitting paradigm, whereby the model memorizes training data rather than learning general patterns.



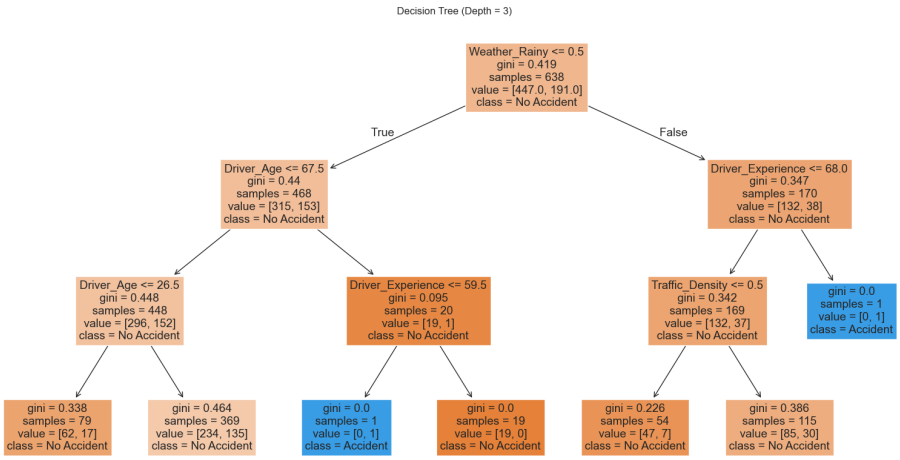
Insight: The best-performing trees strike a balance between too shallow (underfitting) and too deep (overfitting). In this case, a tree depth between 3 and 6 offers a good trade-off.

Tree Visualization (Depth = 3)

This visual shows the internal logic of a decision tree trained with a maximum depth of 3.

Each node splits the dataset based on a feature value for example:

- The first split is on whether the weather is rainy.
- Subsequent decisions depend on Driver_Age, Driver_Experience, and Traffic_Density.

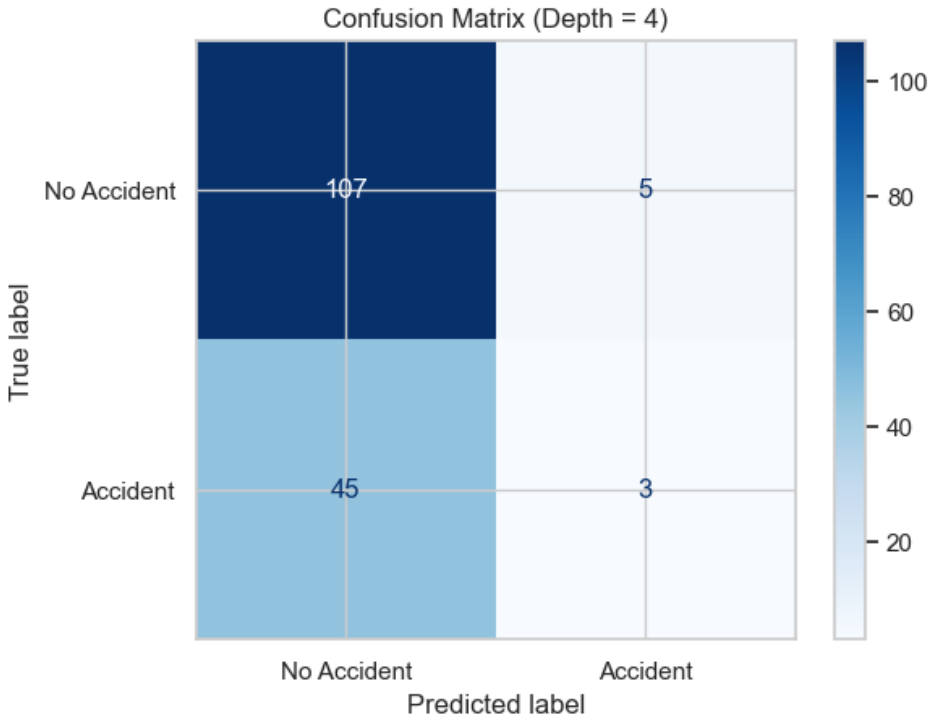


Confusion Matrix (Depth = 4)

The confusion matrix breaks down how well the model predicts each class (Accident vs. No Accident):

	Predicted: No Accident	**Predicted: No Accident**
Actual: No Accident	107	**5**
Actual: Accident	45	**3**

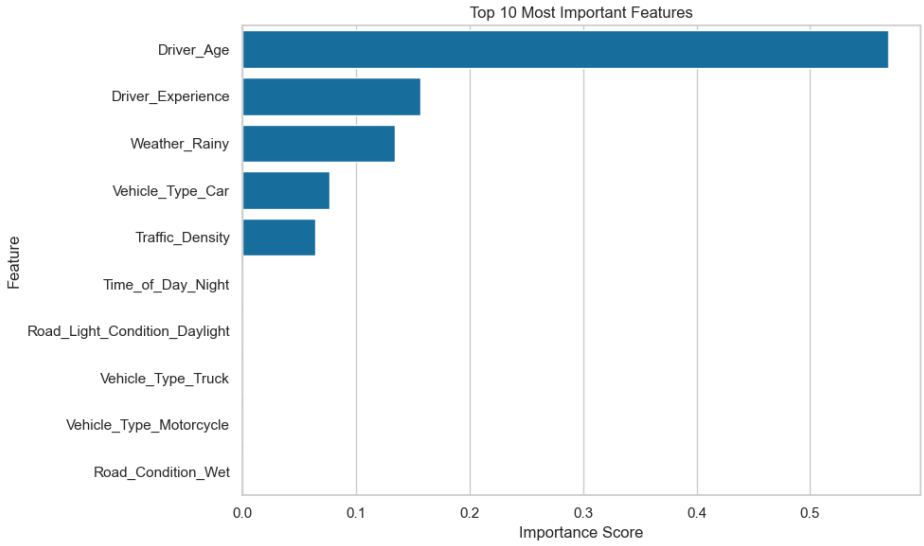
- The model performs well on the majority class (No Accident) but struggles with the minority class (Accident).
- High false negatives (45) suggest the model misses many accident cases an important limitation to address in real-world applications.



Feature Importance

This bar chart highlights the top features the model relies on most during prediction.

- **Driver_Age** is by far the most influential factor, followed by **Driver_Experience** and **Weather_Rainy**.
- Other important features include **Vehicle_Type_Car** and **Traffic_Density**.



Insight: This helps validate that the model focuses on logical and relevant inputs, aligning with real-world intuition an important step in building **trustworthy AI systems**.

Data Preparation and Standardization

Before we train any model, it's important to prepare the data properly so that the model can learn from it effectively and fairly. While Decision Trees don't require feature scaling, we still applied **standardization** to explore its impact and prepare the data for any future experiments with models that do.

We used **StandardScaler**, which transforms each numeric feature by removing its mean and scaling it to unit variance. This step becomes especially important when working with algorithms like KNN, SVM, or XGBoost, where large-scale values could otherwise dominate model behaviour.

Why Standardization Matters

- Ensures all features contribute fairly no single feature outweighs the rest due to scale
- Supports clean integration with other interpretable models or post-hoc explanation methods
- Makes SHAP or tree-based model comparisons more numerically consistent (if used)

Splitting the Dataset

We split our dataset into **70% training** and **30% testing**, using **stratified sampling** to maintain the same ratio of accident and non-accident cases in both sets. This ensures that both subsets represent the true class distribution — a crucial step for fair evaluation.

Conclusion: Understanding Model Decisions with Decision Trees

In this project, we set out to do more than just make predictions. we aimed to **understand them**.

Using a Decision Tree classifier, we explored how machine learning can be applied to something as important and real-world as **predicting traffic accidents**. But unlike many black-box models that just give you a number, Decision Trees help us see *why* a prediction was made.

From the very beginning, we focused on building a model that wasn't just accurate, but also transparent. We started with **exploratory data analysis** to spot patterns and get familiar with the features.

We prepared our data thoughtfully handling missing values, encoding categories, and applying standardization. Then we trained and tested different versions of the model to find the right balance between **performance and simplicity**.

One of the biggest advantages of Decision Trees is that you can read them. You can see that a prediction was made because a driver was young, it was raining, and the road was busy and that's powerful. On top of that, we used **SHAP** to add another layer of insight, showing us exactly how much each feature contributed to each prediction.

References

Bergstra, J., & Bengio, Y. (2012). *Random search for hyper-parameter optimization*. Journal of Machine Learning Research, 13, 281–305. http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

Domingos, P. (2012). *A few useful things to know about machine learning*. Communications of the ACM, 55(10), 78–87. https://doi.org/10.1145/2347736.2347755

Lundberg, S. M., & Lee, S.-I. (2017). *A unified approach to interpreting model predictions*. Advances in Neural Information Processing Systems (NeurIPS), 30. https://papers.nips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., ... & Lee, S.-I. (2020). *From local explanations to global understanding with explainable AI for trees*. Nature Machine Intelligence, 2(1), 56–67. https://doi.org/10.1038/s42256-019-0138-9

Molnar, C. (2022). *Interpretable Machine Learning* (2nd ed.). Leanpub. https://christophm.github.io/interpretable-ml-book/

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). *Scikit-learn: Machine learning in Python*. Journal of Machine Learning Research, 12, 2825–2830. http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

UK Department for Transport. (2022). *Reported road casualties in Great Britain: annual report*. https://www.gov.uk/government/statistics/reported-road-casualties-great-britain-annual-report-2022

Wolberg, W. H., Street, W. N., & Mangasarian, O. L. (1995). *Breast Cancer Wisconsin (Diagnostic) Data Set*. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29