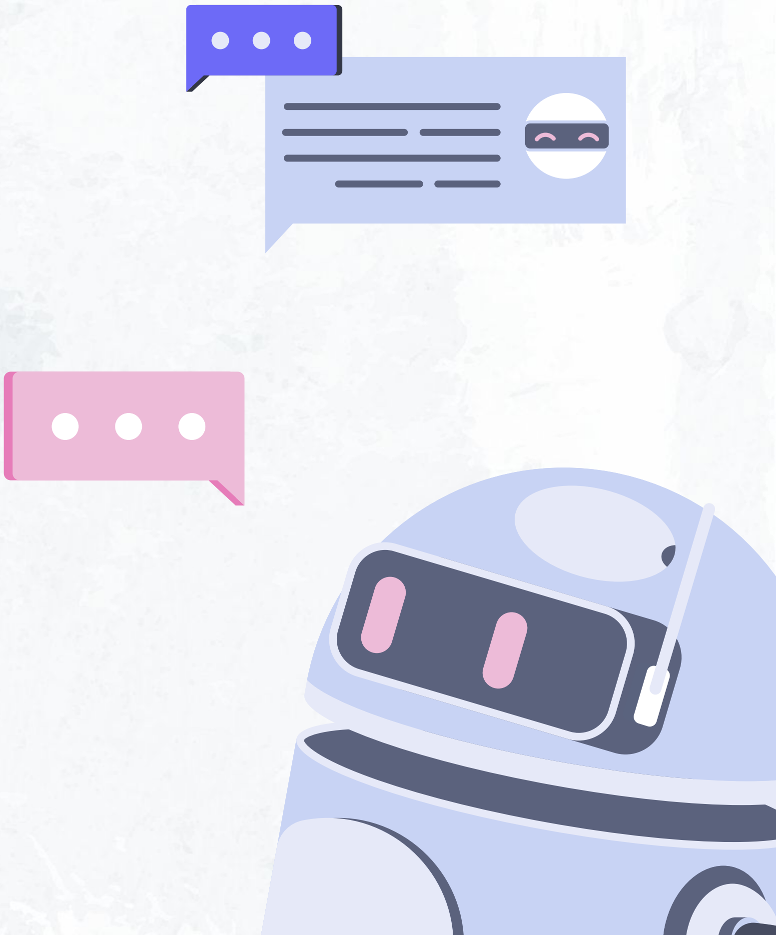
CS638 - Machine Learning (Case Study 2)



Nithish Kumar S (CB.SC.P2CSE24009)

Types of Dataset chosen

01 → Imbalanced Dataset

02 → Non Categorical Dataset

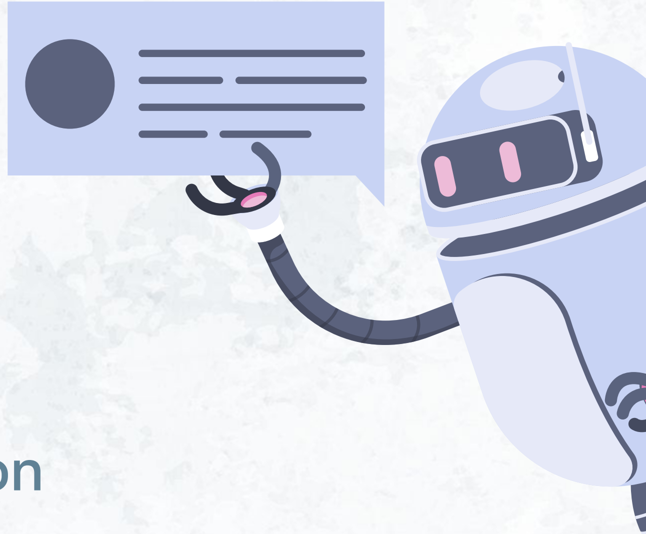
03 → Categorical Dataset

04 → Dataset with many missing values

01

Imbalanced Dataset

Dataset used : Credit Card Fraud Detection



Dataset Overview

Total Records : 284,807 transactions

No of Input features : 30

Output Feature : Class

0 : Legitimate transactions (Non-Fraud)

1 : Fraudulent transactions (Fraud)

Highly Imbalanced Dataset:

- Non-Fraud Cases : **99.83%**
- **Fraud Cases: 0.17%**

How the dataset looks like ?

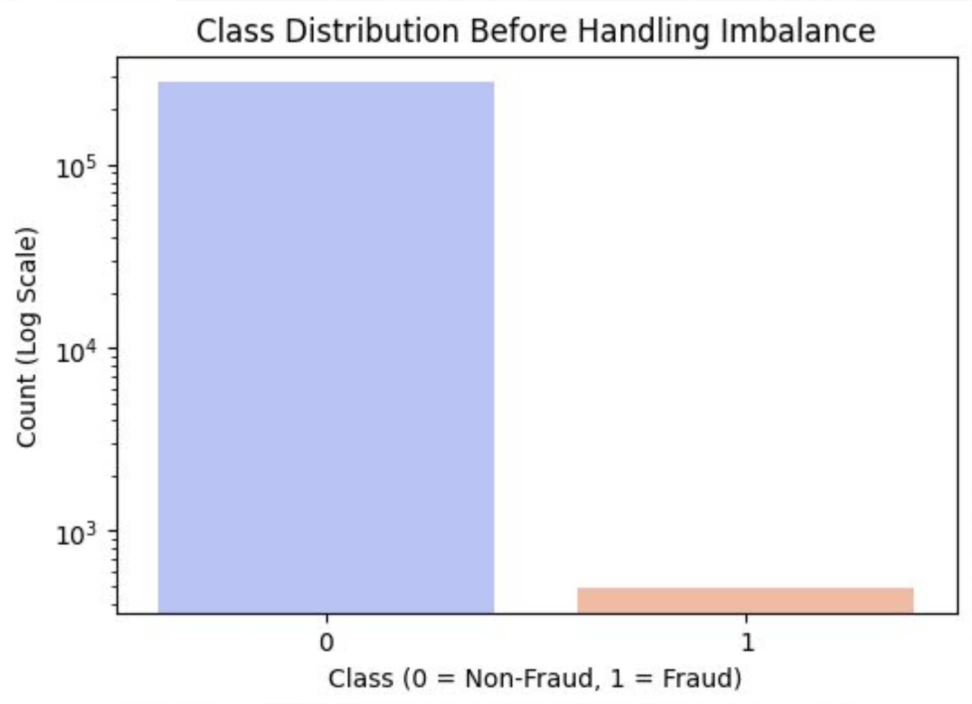
	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

V1 to V28: Principal components extracted using PCA (Feature details are undisclosed due to confidentiality).

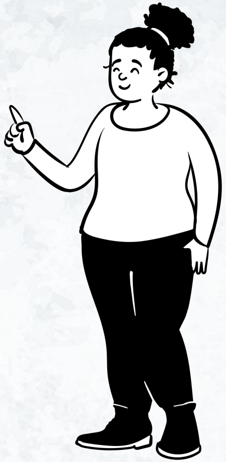
Time: Time elapsed in seconds between the transaction and the first transaction in the dataset.

Amount: Transaction amount in euros.

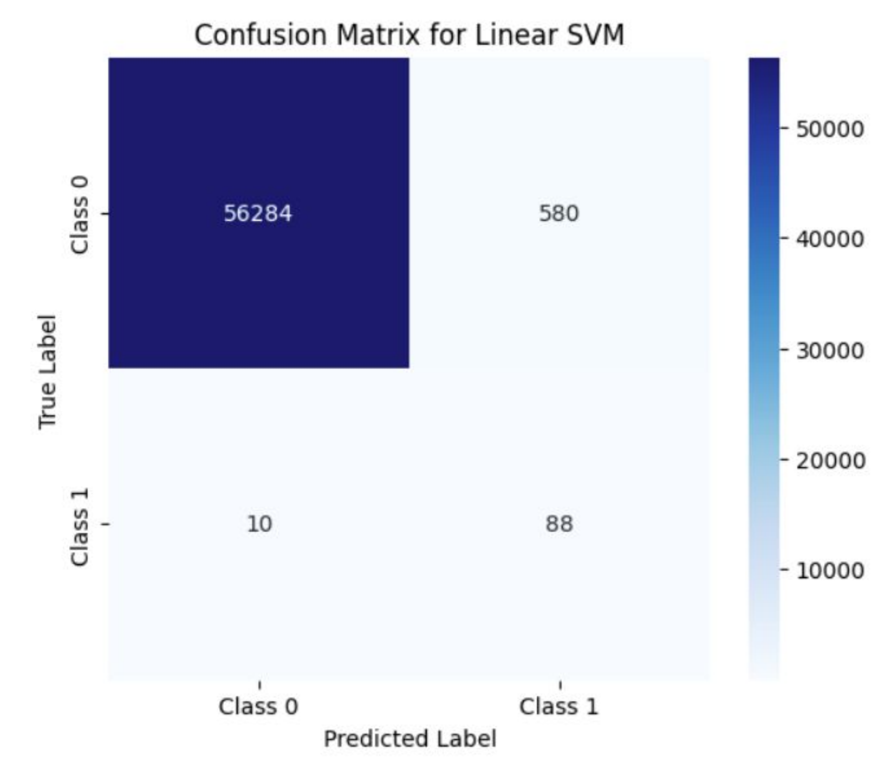
Class Distribution Before Handling Imbalance



Class Distribution After Handling Imbalance



SVM Result Analysis



👉 **Class Imbalance**: Majority of the data belongs to Class 0, making it dominant in predictions.

👉 **High True Negatives (TN = 56,284)**: The model correctly classified most Class 0 instances.

👉 **Moderate False Positives (FP = 580)**: Some Class 0 samples were misclassified as Class 1.

👉 **Low False Negatives (FN = 10)**: Only a few actual Class 1 instances were wrongly predicted as Class 0.

👉 **Improved Recall for Class 1 (TP = 88)**: Compared to earlier results, the model detects more Class 1 instances but still struggles due to class imbalance.

SVM Result Analysis

🔹 Model: Linear SVM

	precision	recall	f1-score	support
0	1.00	0.99	0.99	56864
1	0.13	0.90	0.23	98
accuracy			0.99	56962
macro avg	0.57	0.94	0.61	56962
weighted avg	1.00	0.99	0.99	56962

High accuracy (99%) but may be misleading due to class imbalance.

Class 0 is well-classified with precision, recall, and F1-score close to 1.00.

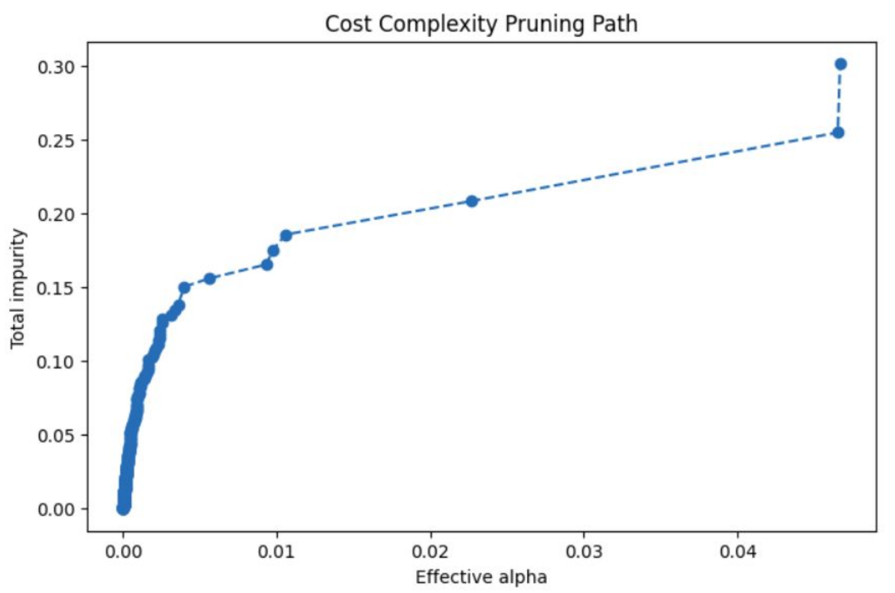
Class 1 suffers from low precision (0.13), indicating many false positives.

Class 1 recall (0.90) is high, meaning most actual positives are detected.

Macro avg F1-score (0.61) is low, highlighting imbalance issues.

Model needs better handling of class imbalance (e.g., oversampling, cost-sensitive learning). 🚀

Cost Complexity Pruning Path Plot



Impurity Increases with Alpha – More pruning leads to higher impurity.

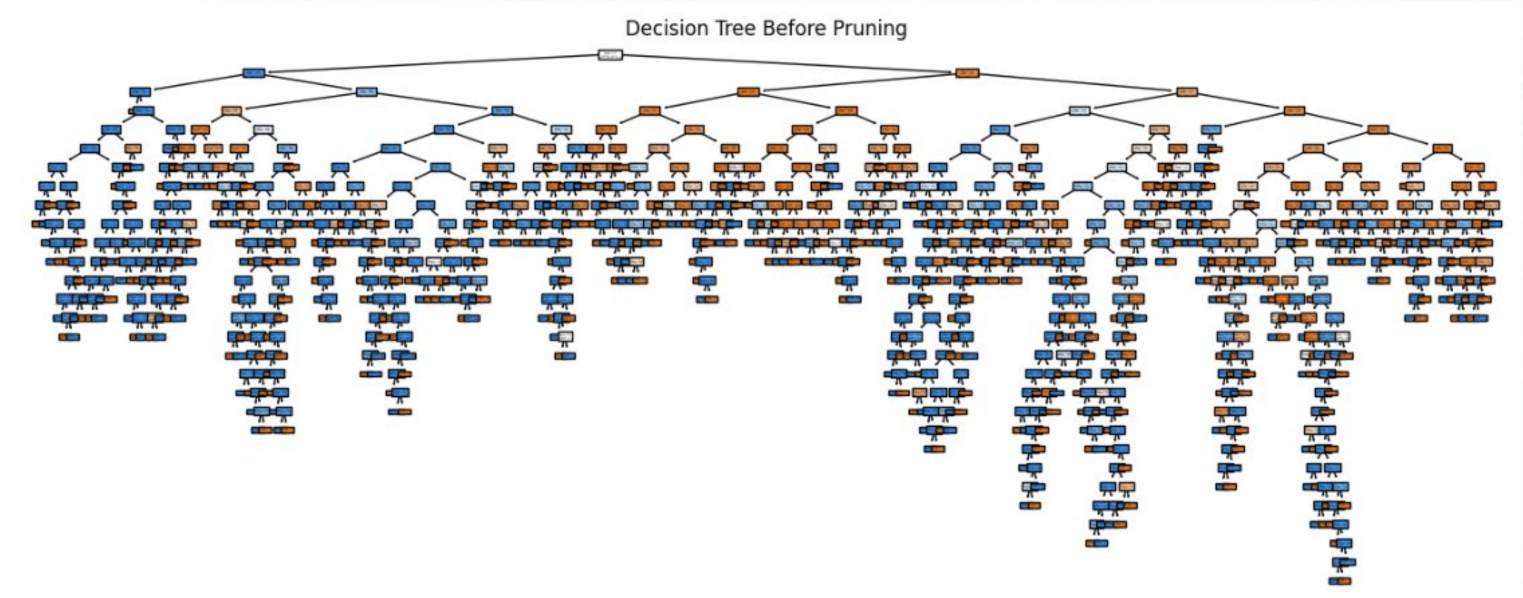
Steep Rise at Low Alpha – Initial pruning removes many small nodes.

Gradual Increase at Higher Alpha – Larger subtrees are pruned, simplifying the model.

Sharp Jump at the End – Indicates heavy pruning, potentially leading to underfitting.

Optimal Alpha Needed – A balance is required to avoid overfitting or underfitting

Decision Tree Before Pruning



Observations

Highly Complex & Overfitted

- The tree is extremely deep with a large number of branches, which suggests overfitting to the training data.
- Overfitting means the model may perform well on training data but generalize poorly to unseen data.

Many Nodes & Splits

- The tree has excessive branching, meaning it is trying to capture too much detail from the dataset, including noise.
- Some splits may be unnecessary and do not contribute significantly to decision-making.

Poor Interpretability

- Due to the large depth and number of nodes, the tree is difficult to interpret and analyze.
- A pruned tree would be more compact and easier to understand.

Observations

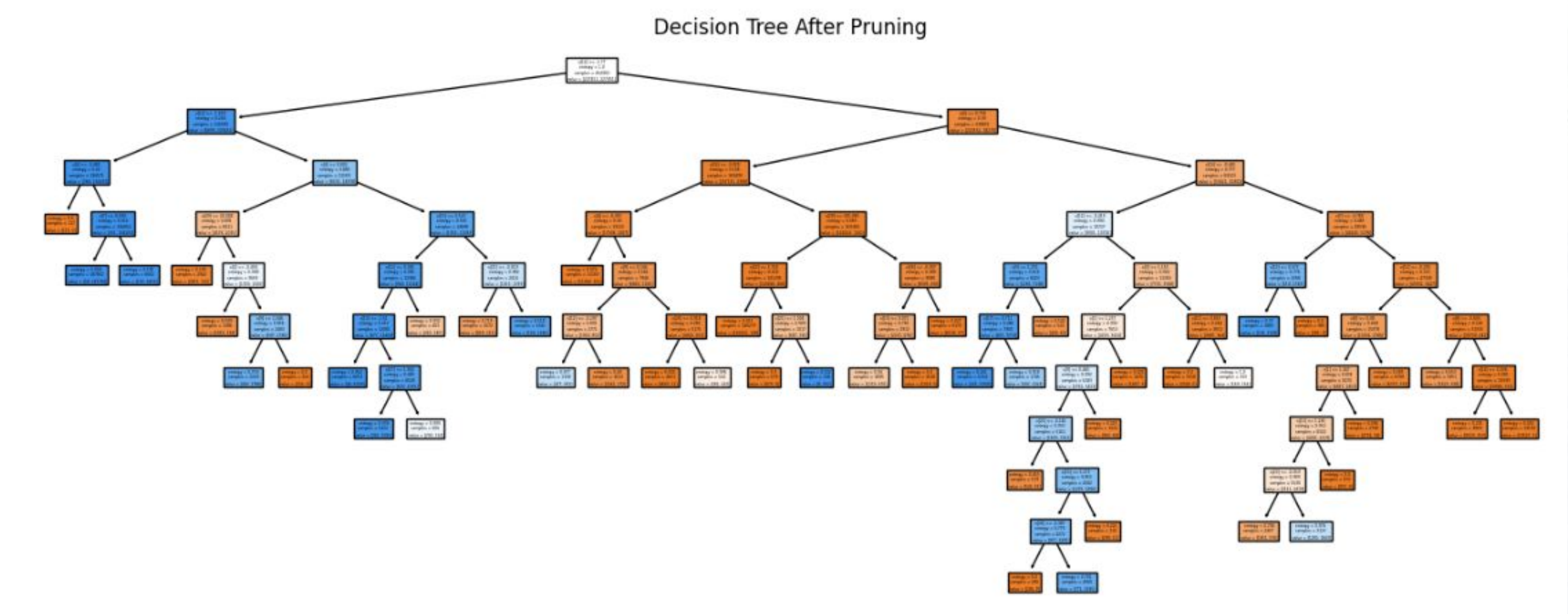
Risk of High Variance

- A deep tree often has high variance, meaning small changes in input data can drastically alter the predictions.
- Pruning can help by simplifying the structure and reducing variance.

Needs Pruning

- The next step should be pruning the tree using **cost complexity pruning (ccp_alpha in scikit-learn)** or setting constraints like `max_depth`, `min_samples_split`, and `min_samples_leaf`.
- This will improve **generalization performance** and prevent overfitting.

Decision Tree After Pruning



Observations

Simplified Structure

- The tree is significantly smaller compared to the unpruned version.
- Many unnecessary branches and splits have been removed, improving interpretability.

Better Generalization

- Pruning reduces overfitting by removing nodes that capture noise in the training data.
- The model is now likely to perform better on unseen data with improved generalization.

Reduced Complexity & Depth

- The depth of the tree has been reduced, making it computationally efficient.
- Shallower trees prevent high variance and increase stability in predictions.

Observations

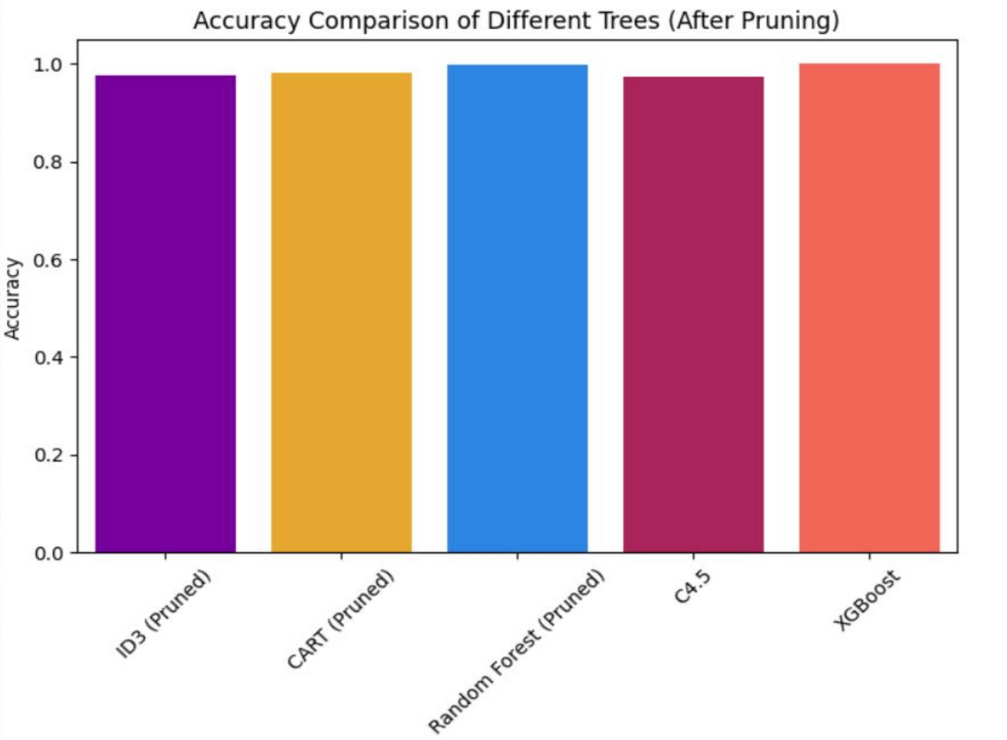
Improved Interpretability

- The tree is now easier to understand and analyze.
- Fewer decision rules make it practical for real-world applications where explainability is important.

Potential Accuracy Trade-Off

- While pruning improves generalization, some accuracy may be lost if too many branches were cut.
- Evaluating post-pruning performance metrics like accuracy, precision, recall, and F1-score is essential.

Model Accuracy Comparison



High Accuracy Across Models – All models achieve near-perfect accuracy after pruning.

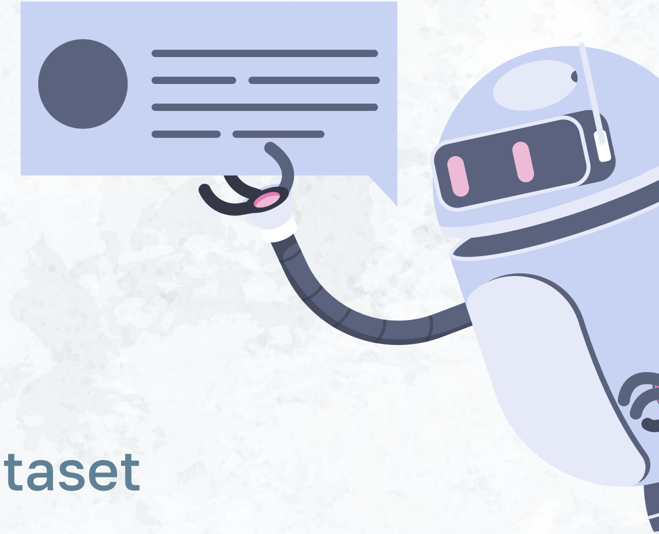
XGBoost Performs Best – It slightly outperforms others, leveraging boosting for better optimization.

Random Forest (Pruned) is Strong – Shows competitive accuracy, benefiting from ensemble learning.

C4.5 Slightly Lower – Performs slightly worse than others but remains highly accurate.

Pruning Maintains Performance – ID3 and CART still perform well, balancing complexity and accuracy.

02 →



Categorical Dataset

Dataset used : Weather Classification Dataset

Dataset Overview

The Weather Classification dataset is designed for predicting different weather conditions based on meteorological parameters. It consists of multiple weather-related features and a target variable representing weather types.

Input features : Temperature, Humidity, Wind Speed, Pressure, Cloud Cover, Visibility, Precipitation

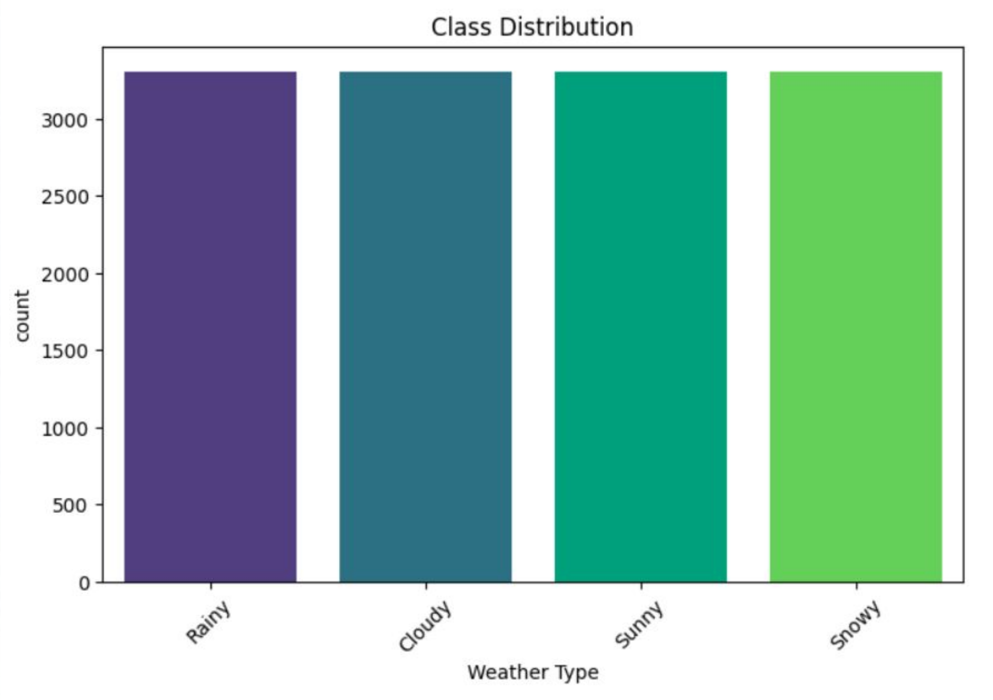
Output Feature : Weather Type

(Sunny - 0, Cloudy - 1, Rainy - 2, Snowy - 3, Stormy - 4)

How the dataset looks like ?

	Temperature	Humidity	Wind Speed	Precipitation (%)	Cloud Cover	Atmospheric Pressure	UV Index	Season	Visibility (km)	Location	Weather Type
0	14.0	73	9.5	82.0	partly cloudy	1010.82	2	Winter	3.5	inland	Rainy
1	39.0	96	8.5	71.0	partly cloudy	1011.43	7	Spring	10.0	inland	Cloudy
2	30.0	64	7.0	16.0	clear	1018.72	5	Spring	5.5	mountain	Sunny
3	38.0	83	1.5	82.0	clear	1026.25	7	Spring	1.0	coastal	Sunny
4	27.0	74	17.0	66.0	overcast	990.67	1	Winter	2.5	mountain	Rainy

Class Distribution



👉 The dataset is **well-balanced**, with nearly equal instances for all weather types.

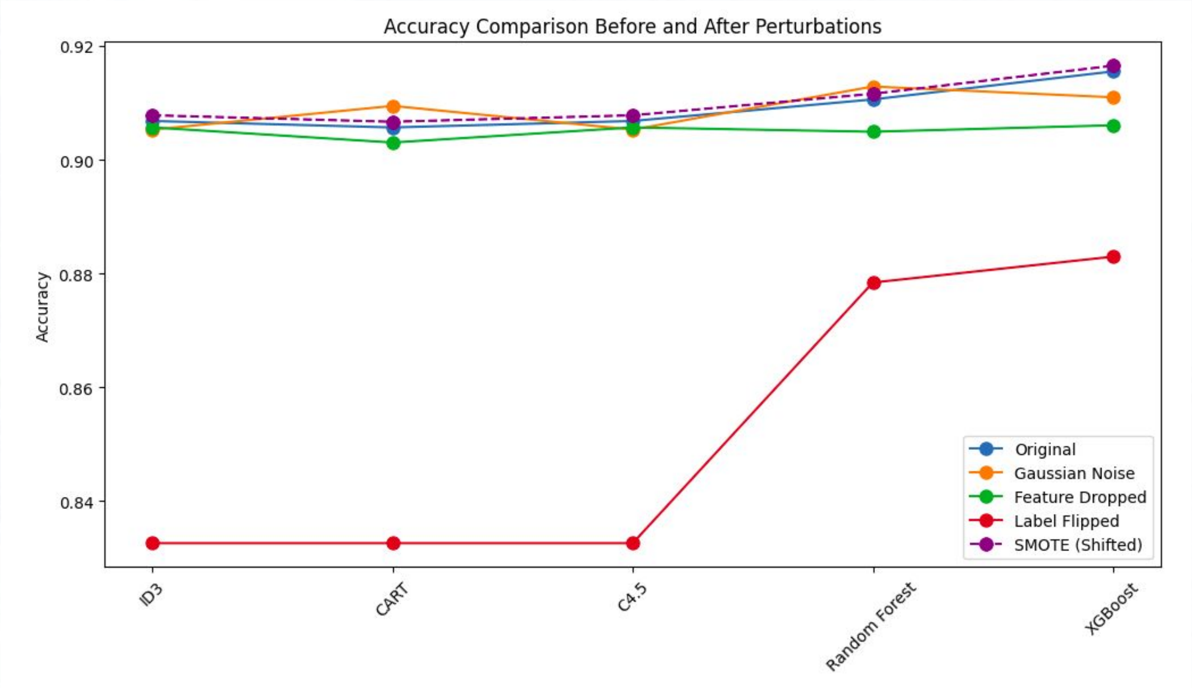
👉 **No significant class imbalance** is observed, reducing the risk of biased model predictions.

👉 Each weather type (Rainy, Cloudy, Sunny, Snowy) has approximately the same count.

👉 This **balanced distribution** ensures that the model **will not favor** any specific weather category.

👉 **A balanced dataset like this is ideal for training classification models without requiring resampling techniques.**

Accuracy Comparison



👉 **Random Forest** and **XGBoost** show the **highest resilience** to perturbations, with minimal accuracy changes.

👉 **ID3** and **CART** models **perform poorly** under **SMOTE** (Shifted), showing significant accuracy drops.

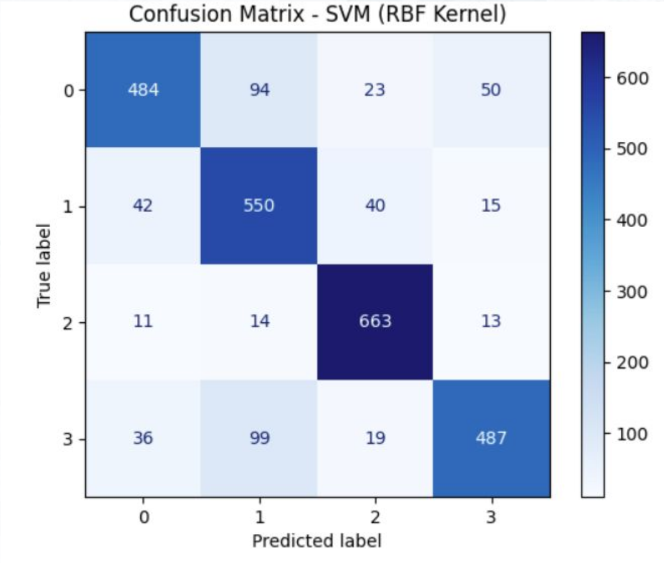
👉 **Gaussian Noise** and **Feature Dropped** perturbations have a negligible impact on accuracy.

👉 **XGBoost consistently achieves the highest accuracy across all perturbations, making it the most robust model.**

Results for SVM

Classification Report for SVM:

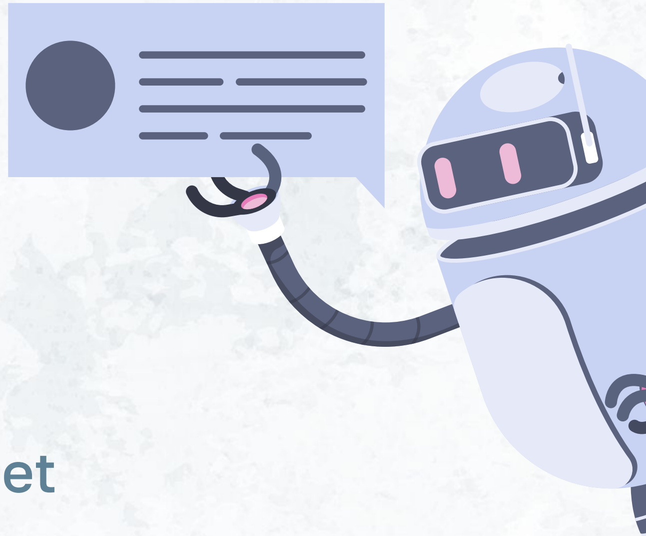
	precision	recall	f1-score	support
0	0.84	0.74	0.79	651
1	0.73	0.85	0.78	647
2	0.89	0.95	0.92	701
3	0.86	0.76	0.81	641
accuracy			0.83	2640
macro avg	0.83	0.82	0.82	2640
weighted avg	0.83	0.83	0.83	2640



03

Non Categorical Dataset

Dataset used : Petrol Consumption Dataset



Dataset Description

Input Features

- **Petrol_tax**: The petrol tax imposed in a particular region.
- **Average_income**: The average income of residents in the region.
- **Paved_highways**: The length or extent of paved highways in the region.
- **Population_driver_licence(%)**: The percentage of the population with a driver's license.

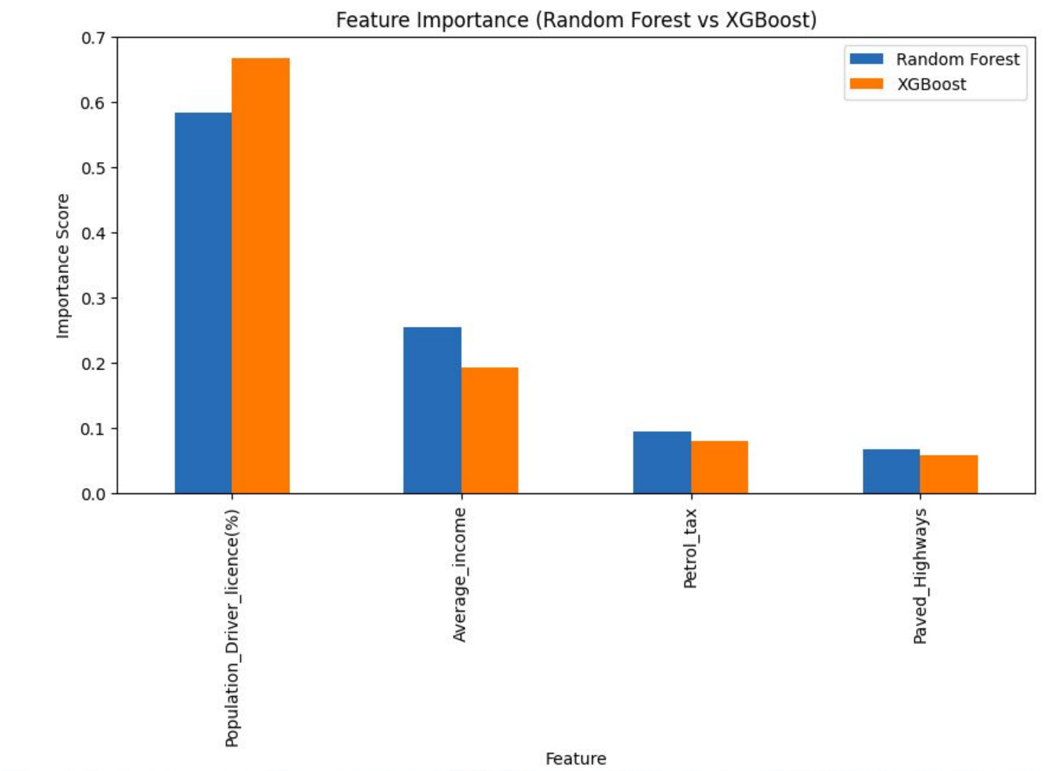
Target Variable

Petrol_Consumption: The amount of petrol consumed, which the dataset aims to predict based on the above features.

How the dataset looks like ?

	Petrol_tax	Average_income	Paved_Highways	Population_Driver_licence(%)	Petrol_Consumption
0	9.0	3571	1976	0.525	541
1	9.0	4092	1250	0.572	524
2	9.0	3865	1586	0.580	561
3	7.5	4870	2351	0.529	414
4	8.0	4399	431	0.544	410

Feature Importance (RF Vs XGBoost)



Population with Driver's License (%) is the most important feature in both models, with XGBoost giving it slightly higher importance.

Average Income is the second most influential factor, with Random Forest weighing it more heavily than XGBoost.

Petrol Tax and **Paved Highways** have the least impact on petrol consumption in both models.

Random Forest and XGBoost show similar patterns, but XGBoost assigns more weight to the top feature.

Feature selection confirms that demographic factors outweigh economic and infrastructure-related ones.

Results Observation

- CART (Squared Error) Performance:
  - MAE: 94.3000
  - MSE: 17347.7000
  - R^2 Score: -1.5858
- C4.5 (Approximation using max_depth=4) Performance:
  - MAE: 96.8000
  - MSE: 16168.1912
  - R^2 Score: -1.4100
- Random Forest Performance:
  - MAE: 53.9610
  - MSE: 6835.4566
  - R^2 Score: -0.0189
- XGBoost Performance:
  - MAE: 73.9795
  - MSE: 12462.1939
  - R^2 Score: -0.8576
- SVM (RBF Kernel) Performance:
  - MAE: 64.0571
  - MSE: 6724.8329
  - R^2 Score: -0.0024

👉 Random Forest has the lowest MAE (53.96) and MSE (6835.45), making it the best-performing model.

👉 CART and C4.5 perform poorly with high MAE and MSE, indicating weaker predictive power.

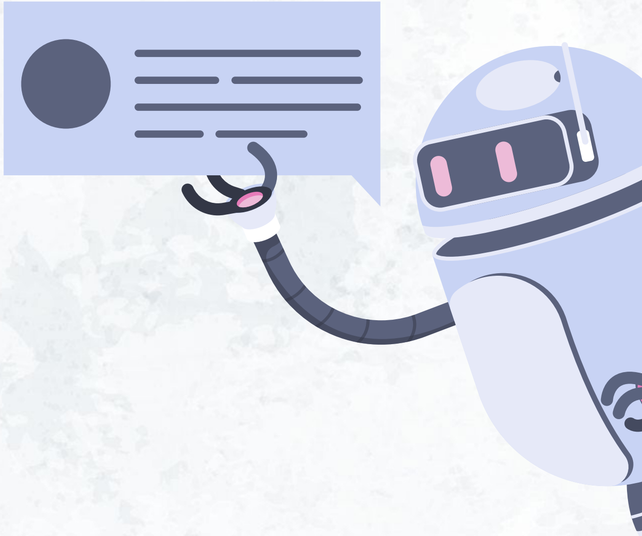
👉 XGBoost underperforms compared to Random Forest, with a higher MAE (73.97) and negative R^2 (-0.8576).

👉 SVM achieves a relatively low MAE (64.06) and the least negative R^2 (-0.0024), suggesting decent generalization.

04 →

Missing value Dataset

Dataset used : Titanic Dataset



Dataset Description

The Titanic dataset is a well-known dataset used for machine learning and data analysis, particularly for classification problems. It contains information about passengers on the Titanic and whether they survived or not.

Input Features : PassengerId, PClass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked

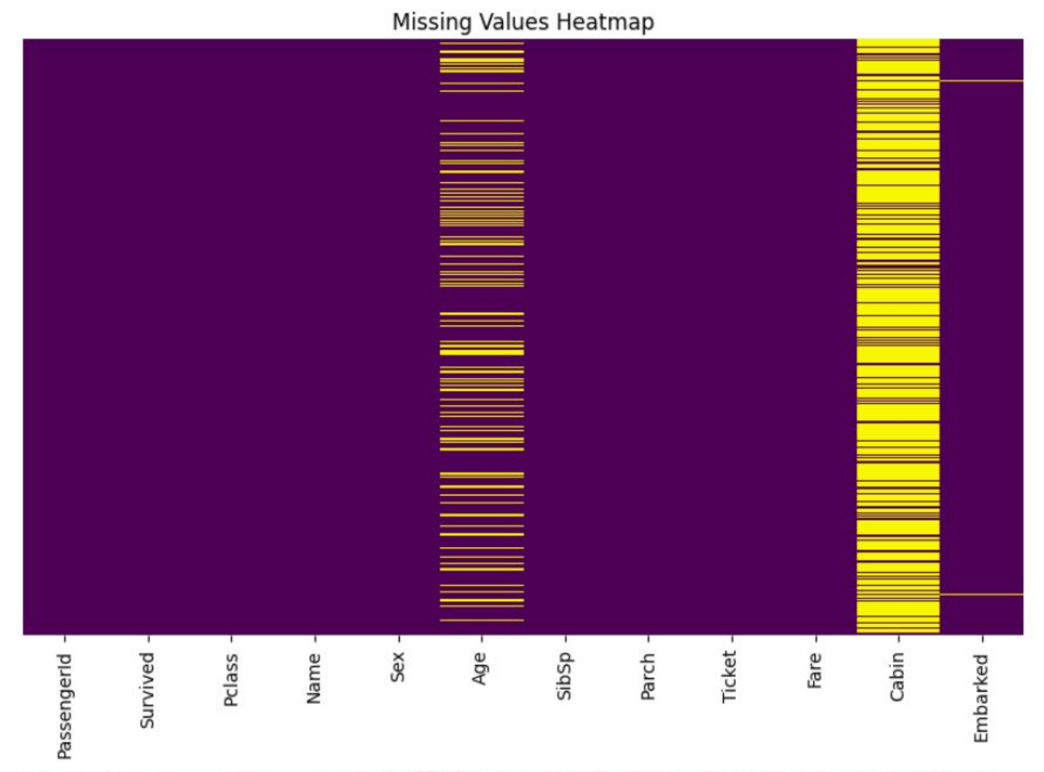
Output Feature : Survived

0 - No (Did not survive)
1 - Yes (Survived)

How the dataset looks like ?

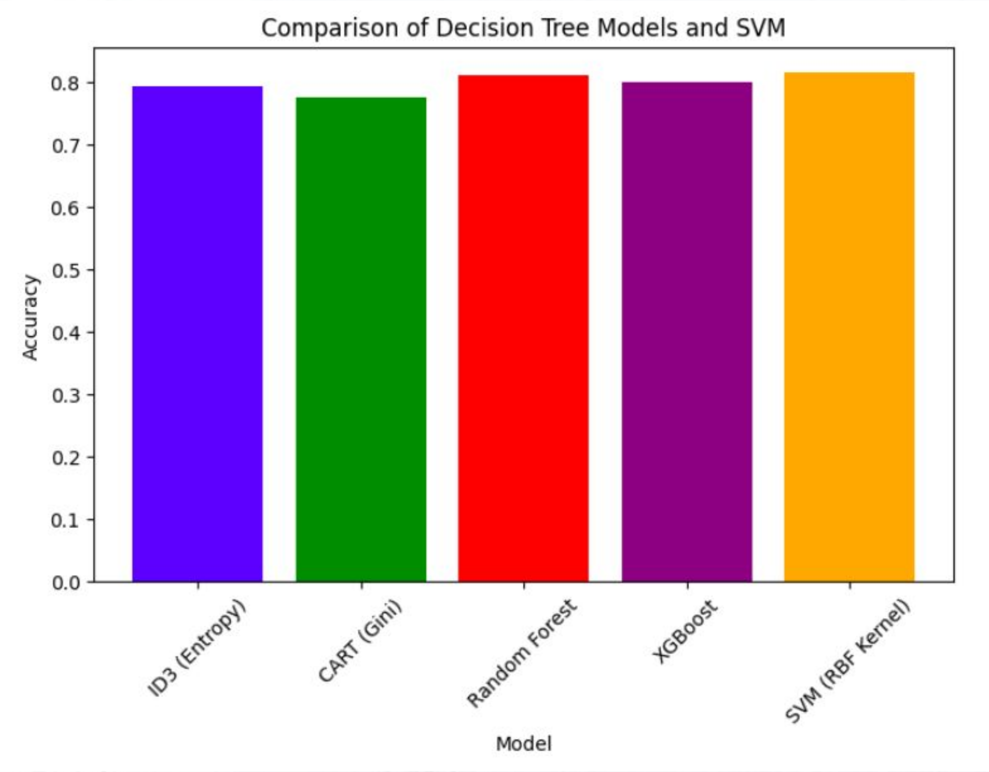
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Missing Values Visualization



- Cabin has the most missing values
- Age Has Some Missing Values
- Embarked Has a Few Missing Values
- Other Columns Have No Missing Values

Model Performance Comparison



Random Forest achieves the highest accuracy, slightly above 0.8, indicating strong performance.

SVM (RBF Kernel) also performs well, showing the effectiveness of non-linear decision boundaries.

XGBoost is competitive, proving the advantage of gradient boosting over single decision trees.

ID3 (Entropy) and CART (Gini) have the lowest accuracy, suggesting weaker generalization.

Ensemble methods (Random Forest, XGBoost) outperform basic decision trees, reducing overfitting.

Thank You!