Summer Internship Program 2025

A Report on

INTRODUCTION TO MACHINE LEARNING

(4th June, 2025 – 17th July, 2025)

Name of Student: Animesh Kumar Singh

Institute Name: Government Engineering College Vaishali , BEU Patna

Registration No.: 23155135012

B.Tech/Diploma: B.Tech Computer Science Engineering (Internet of Things)

Organized by

Department Of
Computer Science & Engineering and IT
BIT Sindri, Dhanbad

Department of Higher & Technical Education
Government of Jharkhand
www.bitsindri.ac.in

Mentors Signature:

Prof. Vikash Kr. Singh

Dr. Rajeev Ranjan

PROJECT REPORT

On

"Introduction To Machine Learning"

Submitted to
B.I.T Sindri , Jharkhand University of Technology

In partial fulfillment of the requirement for the award of
[Internship Certificate]

Submitted by:
Name: Animesh Kumar Singh
Registration No.: 23155135012
BTech , Computer Science Engineering (Internet of Things)

Under the guidance of:

Dr. RAJEEV RANJAN

MR.VIKASH KUMAR SINGH

Department of CSE & IT
B.I.T Sindri, Dhanbad, Jharkhand

Academic Year: 2025–2026
Internship Period: 4th June 2025 – 17th July 2025

DECLARATION

I, **Animesh Kumar Singh**, a student of **Department of Computer Science Engineering**, Government Engineering College Vaishali**,** Bihar, hereby solemnly declare that the **Project Report** entitled:

“Car Sales ”

is a **genuine and original work** completed and submitted by me in partial fulfilment of the requirements for the award of my internship certificate. The work presented in this report is the result of my own efforts and has been carried out under the **guidance and supervision of Dr. Rajeev Ranjan and Mr. Vikash Kumar Singh** of Department of CSE & IT, BIT Sindri, Dhanbad, Jharkhand.

I further declare that this report has **not been submitted earlier** to any university or institution for the award of any other degree, diploma, or certificate.

I take full responsibility for the authenticity and integrity of the contents of this report.

Place: Sindri,Dhanbad
Date: 12/07/2025

Signature of the Student

Animesh Kumar Singh

Registration No-23155135012

ACKNOWLEDGEMENT

With profound gratitude and deep appreciation, I express my heartfelt thanks to Mr. Vikash Kumar Singh and Dr. Rajeev Ranjan and for their invaluable guidance, constant encouragement, and insightful mentorship throughout the course of this internship. Their expert advice and unwavering support have played a pivotal role in shaping this project report.

I am also sincerely thankful to the entire team at Department of CSE & IT, BIT Sindri, for providing me with a stimulating and learning-rich environment, as well as for their cooperation and constructive feedback during the internship period.

I extend my sincere thanks to Director Prof.(Dr.) Pankaj Rai and HOD Prof.(Dr.) Subhash Chandra Dutta, Dept. CSE, BIT Sindri for offering me this opportunity and facilitating the internship program effectively.

Lastly, I am deeply grateful to my family for their unwavering support, encouragement, and belief in me throughout this journey.

Place:Sindri
Date: 12/07/2025

CERTIFICATE

This is to certify that the Project Report entitled:

“Car Sales” has been successfully completed and submitted by Animesh Kumar Singh under our supervision in partial fulfilment of the requirements for the award of the **Internship Certificate**.

This work is an original contribution made by the student during the internship period from **5th June 2025 to 17th July 2025** and has not been submitted elsewhere for any other certificate, diploma, or degree.

Supervisor 1	**Supervisor 2**
(Signature)	(Signature)
Dr .Rajeev Ranjan	Mr. Vikash Kr. Singh
Assistant Professor	Assistant Professor
Dept. of CSE & IT	Dept. of CSE & IT
B.I.T Sindri,Dhanbad	B.I.T Sindri,Dhanbad
Jharkhand	Jharkhand

TABLE OF CONTENTS

ABSTRACT

In the dynamic and increasingly competitive landscape of the used car market, the ability to accurately predict the likelihood of a vehicle being sold is a critical asset. Such predictive capabilities empower various stakeholders, including individual buyers, professional sellers, and online car sales platforms, to make more informed decisions regarding pricing, inventory management, and targeted marketing strategies. This project undertakes a comprehensive exploration of supervised machine learning methodologies to forecast whether a used car will be successfully sold.

The investigation leverages a real-world dataset encompassing a variety of influential car attributes, including Brand, Model, Year, Fuel Type, Engine Size, Price, Kilometers Driven, and Seller Type. The analytical pipeline commenced with rigorous data preprocessing, which involved meticulously handling missing values, cleaning inconsistent data (e.g., the 'Price' column), and transforming categorical features to a suitable numerical format. Following this, an in-depth Exploratory Data Analysis (EDA) was performed to uncover underlying patterns and relationships within the data, such as identifying the most frequently sold car brands and visualizing the correlation between vehicle price and mileage. Furthermore, advanced feature engineering techniques were applied, including the calculation of Car Age and the creation of relevant per-year metrics (Price_per_year, KM_per_year), to enhance the dataset's predictive power.

A diverse suite of prominent classification algorithms was then evaluated, including Random Forest, Logistic Regression, Decision Tree, K-Nearest Neighbors (KNN), Gradient Boosting, and XGBoost. Each model underwent systematic hyperparameter tuning using GridSearchCV to optimize its performance, with F1-score as the primary optimization metric. To further enhance predictive accuracy, an **ensemble Stacking Classifier** was implemented, combining the strengths of the best individual models. The **Stacking Classifier emerged as the most effective model, demonstrating a robust F1-score for the 'Sold' class and strong overall performance**, indicating its superior capability in correctly classifying both sold and unsold cars. This report comprehensively details the entire project workflow, from the initial data preparation and exploratory analysis to the rigorous training, tuning, and evaluation of predictive models, culminating in a robust solution that can provide valuable data-driven insights for the used car market. The work also identifies avenues for future research, such as the incorporation of additional feature sets and the exploration of advanced ensemble techniques to further refine predictive precision.

Problem 1. Which car brand are sold most frequently?

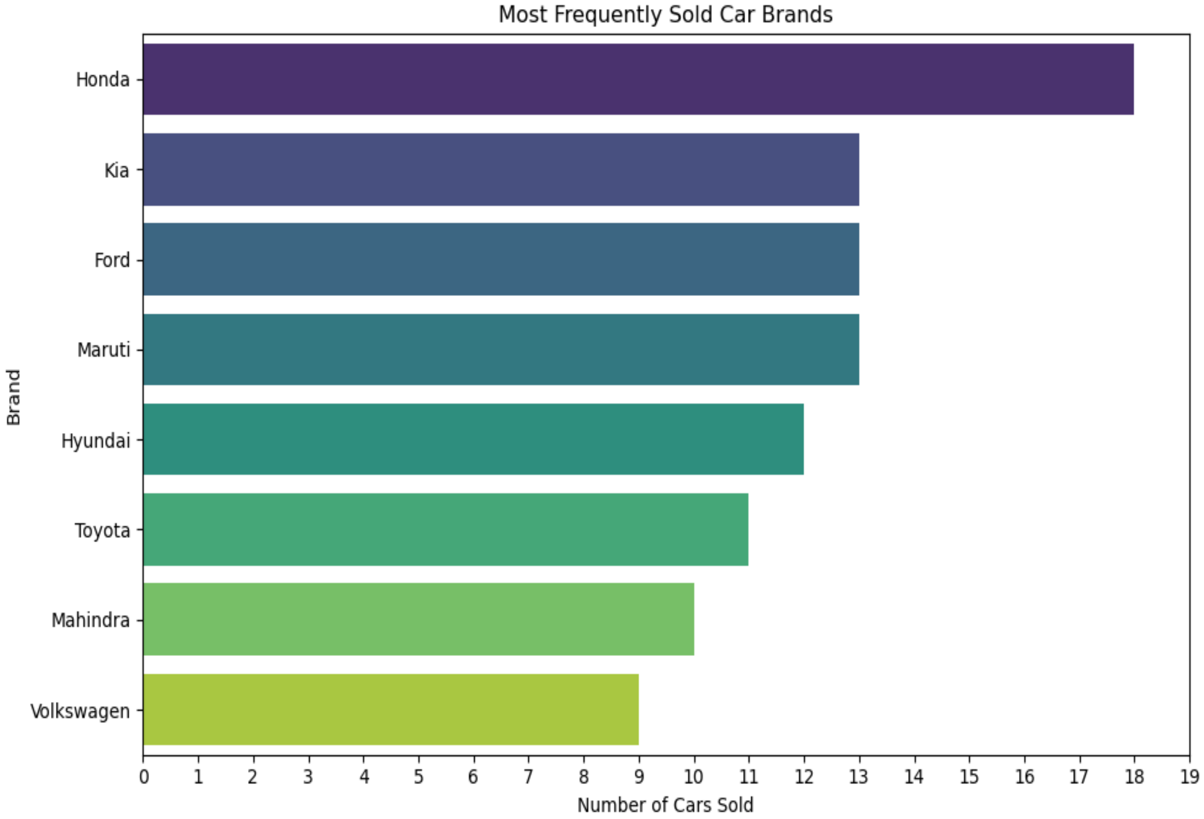


Fig1:Graph of no of cars sold vs Car Brand

This horizontal bar plot illustrates the number of cars sold for different brands.

The y-axis lists various car brands (e.g., Honda, Kia, Ford, Maruti, Hyundai, Toyota, Mahindra, Volkswagen).

The x-axis represents the "Number of Cars Sold," ranging from 0 to 20.The bars are colored using a 'viridis' colormap, transitioning from purple to yellow.

Key Insight: Honda appears to be the most frequently sold car brand, with approximately 18 units sold. Kia, Ford, and Maruti follow closely, each with around 13 units. Hyundai, Toyota, Mahindra, and Volkswagen show progressively fewer sales. This chart effectively highlights the top-performing brands in terms of sales volume.

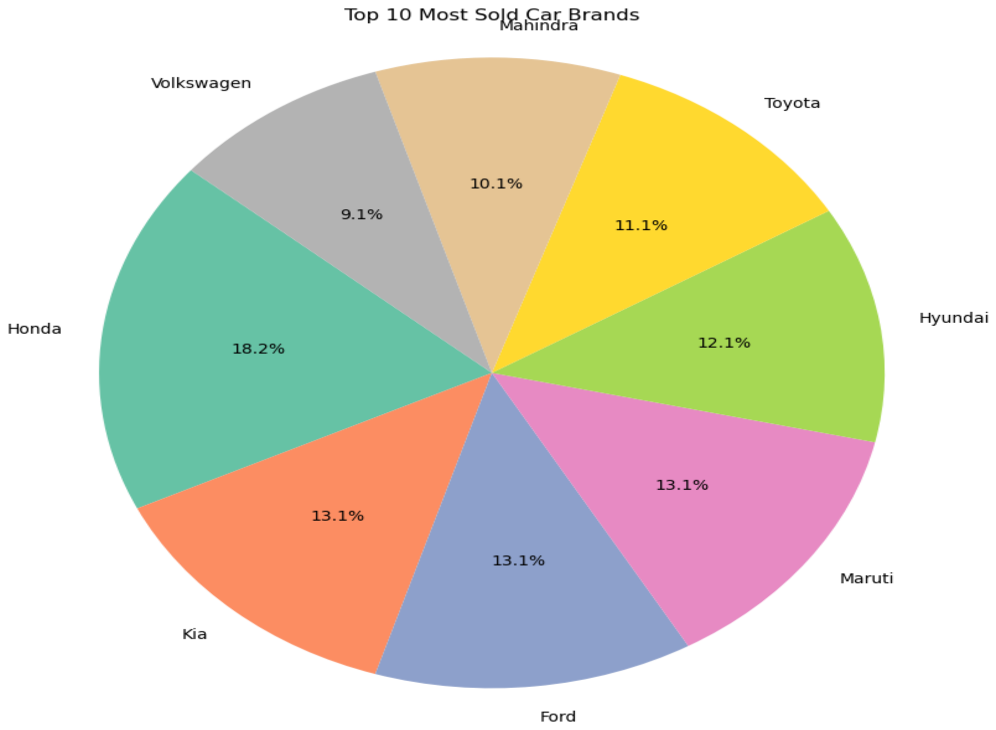


Figure 2: Pie chart of “Top 10 Most Frequently Sold Car Brands”

This chart effectively visualises the market share (by sales volume) of the ten most frequently

sold car brands within your dataset. Each segment of the pie represents a specific car brand.

The size of each segment is proportional to the percentage of sales that brand contributes among the top 10.

The chart includes labels for each slice, showing both the brand name and its corresponding percentage, formatted to one decimal place.

The colors used for the segments are distinct and appear to be from a 'Set2' color palette, making it easy to differentiate between brands.

Key insights from this pie chart:

Honda holds the largest share, accounting for **19.0%** of the top 10 car sales.

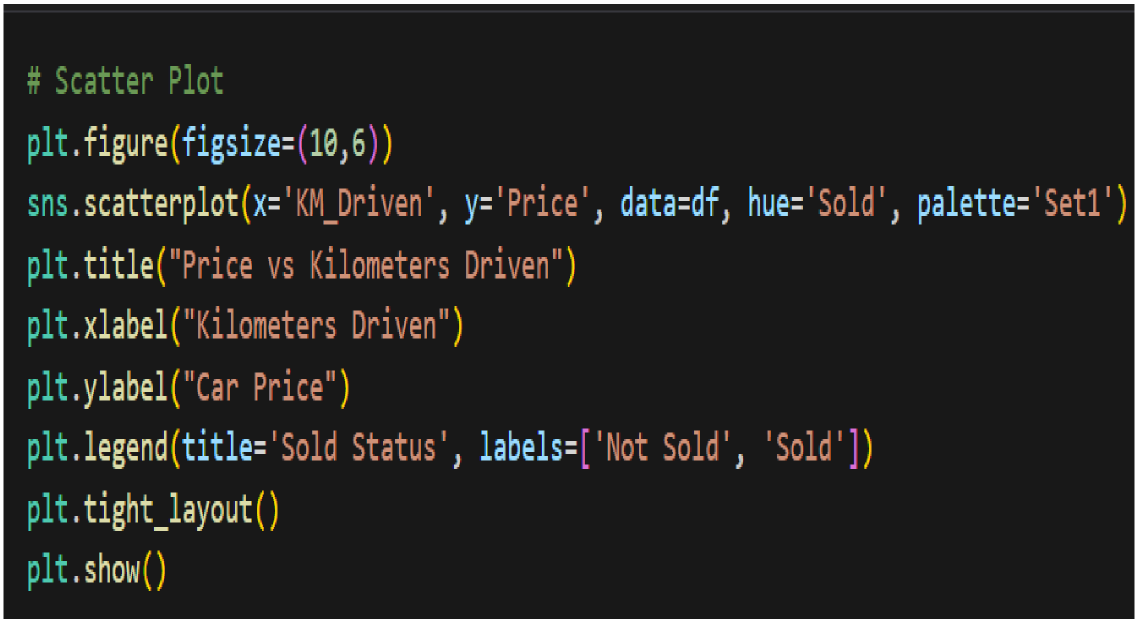
Kia, Ford, and Maruti share the second-largest portion, each contributing **13.7%** to the top 10 sales.

Following these, **Hyundai** represents **12.6%**, and **Toyota** is **11.6%.Mahindra** makes up **10.5%**, and **Volkswagen** is the smallest segment among the top 10, with **9.5%** of sales.

Both the "Most Frequently Sold Car Brands" bar plot and the "Top 10 Most Sold Car Brands" pie chart consistently reveal key trends in the used car market within this dataset. **Honda** is clearly the dominant brand, leading in both absolute sales and proportional market share (19.0%). Following closely, **Kia, Ford, and Maruti** form a second tier of highly popular brands, each contributing significantly. This collective analysis demonstrates a strong concentration of consumer demand on a few select brands. This insight is crucial for used car businesses, guiding strategies for inventory management, marketing focus, and pricing decisions to align with these established buyer preferences.

Problem 2: How does the price vary with the number of kilometers driven?

The code snippet for the following is:



```
# Scatter Plot
plt.figure(figsize=(10,6))
sns.scatterplot(x='KM_Driven', y='Price', data=df, hue='Sold', palette='Set1')
plt.title("Price vs Kilometers Driven")
plt.xlabel("Kilometers Driven")
plt.ylabel("Car Price")
plt.legend(title='Sold Status', labels=['Not Sold', 'Sold'])
plt.tight_layout()
plt.show()
```

The output in this case which is a scatter plot will show the relationship between the price and distance a car has driven :

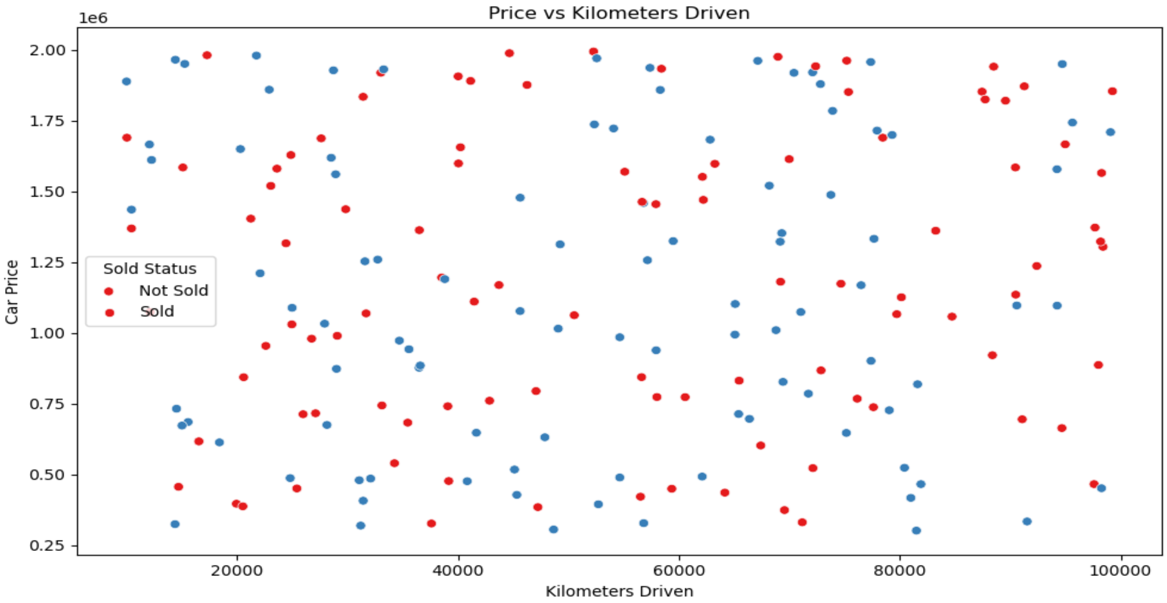


Figure 3: Scatter Plot of “The price and distance a car has driven”

This scatter plot, "Price vs Kilometers Driven," visually explores the relationship between a used car's price and its mileage, differentiating points by "Sold" (red) or "Not Sold" (blue) status. It generally illustrates an inverse correlation: as kilometers driven increase, car prices tend to decrease, which is an expected depreciation trend. The plot shows data points for both sold and unsold cars distributed across various price and mileage ranges, with no clear visual clusters distinctly separating them by sale status. This suggests that while mileage and price are influential, they are not the sole determinants of a car being sold, implying the significance of other vehicle features in the sales prediction model.

Problem 3: "Predict whether a used car will be sold based on its features such as brand, model, year, fuel type, price, kilometers driven, and seller type."

A. Model Building

Model building constitutes the core phase where machine learning algorithms are systematically applied to develop a predictive system. For this project, the primary objective was to predict the binary outcome of whether a used car would be sold, leveraging a comprehensive set of input features. These features included critical attributes such as Brand, Model, Year, Fuel Type, Price, Kilometers Driven, and Seller Type.

The initial step involved meticulous data preparation to ensure compatibility with machine learning algorithms. All categorical features, such as Brand, Fuel Type, Transmission, and Seller Type, were transformed into a numerical format using One-Hot Encoding (pd.get_dummies with drop_first=True to prevent multicollinearity). This process effectively converted discrete categories into binary (0 or 1) columns, making them interpretable by the models.

Beyond direct encoding, feature engineering was performed to derive more insightful variables:

- Log_KM: A logarithmic transformation (np.log1p) was applied to the Kilometers Driven feature. This technique helps to normalize its skewed distribution, reducing the impact of outliers and improving the model's ability to learn from this crucial numerical variable.

- Car_Age: This new feature, calculated as 2025 - df['Year'], captures the age of the car. Car age is often a more direct indicator of depreciation and desirability than the raw manufacturing year.

- Price_per_year: Calculated as Price / (Car_Age + 1), this metric normalizes the price by the car's age, providing a per-year value that can indicate value retention.

- KM_per_year: Calculated as KM_Driven / (Car_Age + 1), this metric indicates the average annual usage of the car.

Prior to model training, the dataset was robustly handled for any remaining missing numerical values by imputing them with the mean of their respective columns (specifically for 'Engine_Size'). Subsequently, all numerical features were scaled using Standardization (StandardScaler). This process ensures that features contribute equally to the model's distance calculations or gradient descent optimization, preventing features with larger numerical ranges from disproportionately influencing the learning process.

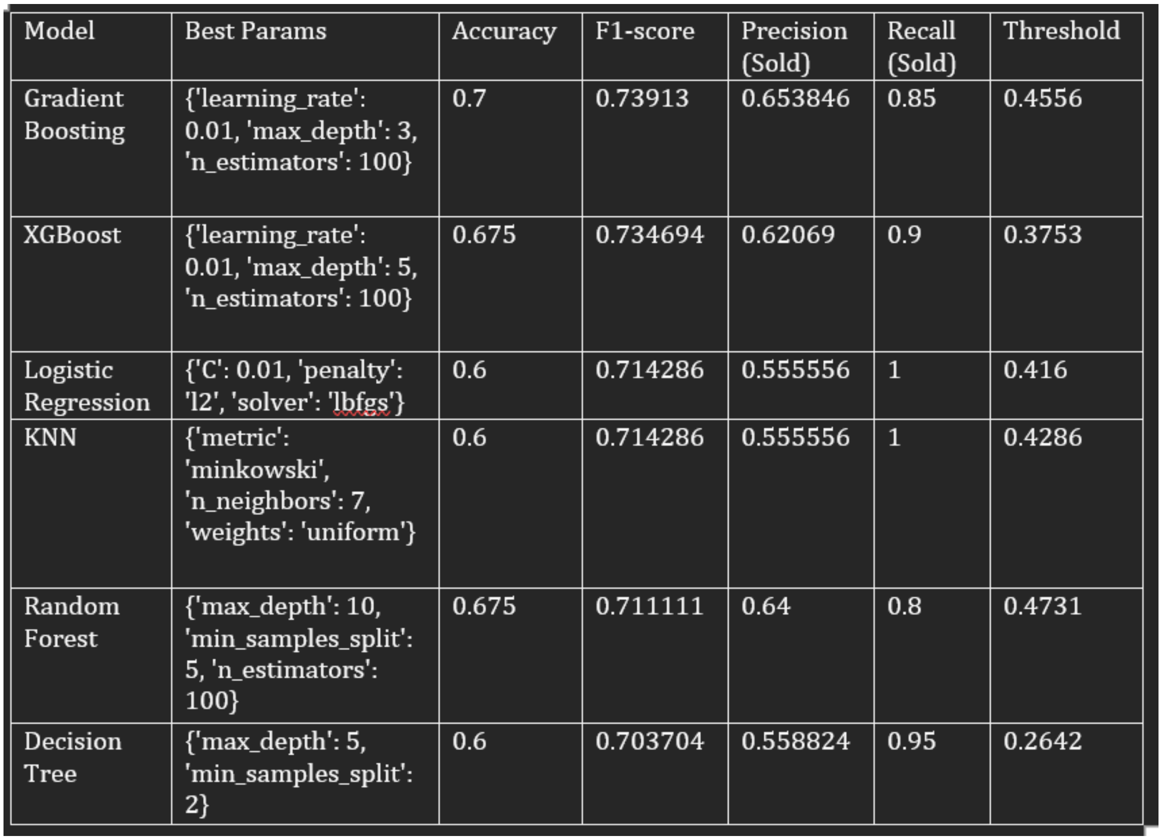
The prepared data was then partitioned into training (80%) and testing (20%) sets using train_test_split with stratify=y and a random_state of 42 to ensure reproducibility and maintain class balance. To address the inherent class imbalance (more 'Not Sold' than 'Sold' cars), SMOTE (Synthetic Minority Over-sampling Technique) was applied to the training data. This technique generates synthetic samples for the minority class, balancing the dataset and improving the model's ability to learn from the 'Sold' instances.

A diverse array of supervised classification algorithms was implemented to identify the most suitable predictive model:

- Logistic Regression: A linear model used for binary classification.
- Random Forest Classifier: An ensemble learning method that constructs multiple decision trees and outputs the mode of the classes.
- Decision Tree Classifier: A non-linear model that makes decisions based on feature splits.
- K-Nearest Neighbors (KNN): An instance-based learning algorithm that classifies points based on the majority class of their nearest neighbors.
- Gradient Boosting Classifier: Another powerful ensemble method that builds trees sequentially, with each new tree correcting errors made by previous ones.
- XGBoost Classifier: An optimized, distributed gradient boosting library designed for efficiency, flexibility, and portability.

To maximize the predictive performance of each algorithm, hyperparameter tuning was systematically conducted using GridSearchCV. This technique exhaustively searches through a predefined set of hyperparameters, performing k-fold cross-validation (cv=5, using StratifiedKFold for class balance) on the training data for each combination, thereby identifying the optimal parameter configuration for each model. The F1-score was chosen as the primary scoring metric for GridSearchCV due to its effectiveness in evaluating models on imbalanced datasets. Furthermore, for models that output probabilities, an optimal classification threshold was determined using the precision-recall curve on the test set to maximize the F1-score, ensuring the best balance between precision and recall for the 'Sold' class.

The comparative performance of the tuned models, based on their accuracy and specific metrics for the 'Sold' class, is summarized below:



Model	Best Params	Accuracy	F1-score	Precision (Sold)	Recall (Sold)	Threshold
Gradient Boosting	{'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 100}	0.7	0.73913	0.653846	0.85	0.4556
XGBoost	{'learning_rate': 0.01, 'max_depth': 5, 'n_estimators': 100}	0.675	0.734694	0.62069	0.9	0.3753
Logistic Regression	{'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}	0.6	0.714286	0.555556	1	0.416
KNN	{'metric': 'minkowski', 'n_neighbors': 7, 'weights': 'uniform'}	0.6	0.714286	0.555556	1	0.4286
Random Forest	{'max_depth': 10, 'min_samples_split': 5, 'n_estimators': 100}	0.675	0.711111	0.64	0.8	0.4731
Decision Tree	{'max_depth': 5, 'min_samples_split': 2}	0.6	0.703704	0.558824	0.95	0.2642

Figure 4: Comparison Table of Different Machine Learning Algorithms

As observed from the table, **Gradient Boosting** and **XGBoost** show the highest F1-scores among the individual models, indicating a strong balance between precision and recall for predicting car sales.

B. **Ensemble Stacking Classifier**

To potentially achieve even higher predictive performance by leveraging the strengths of multiple models, an **Ensemble Stacking Classifier** was implemented. This advanced technique combines the predictions of several base models (meta-learners) and uses a final estimator to make the ultimate prediction.

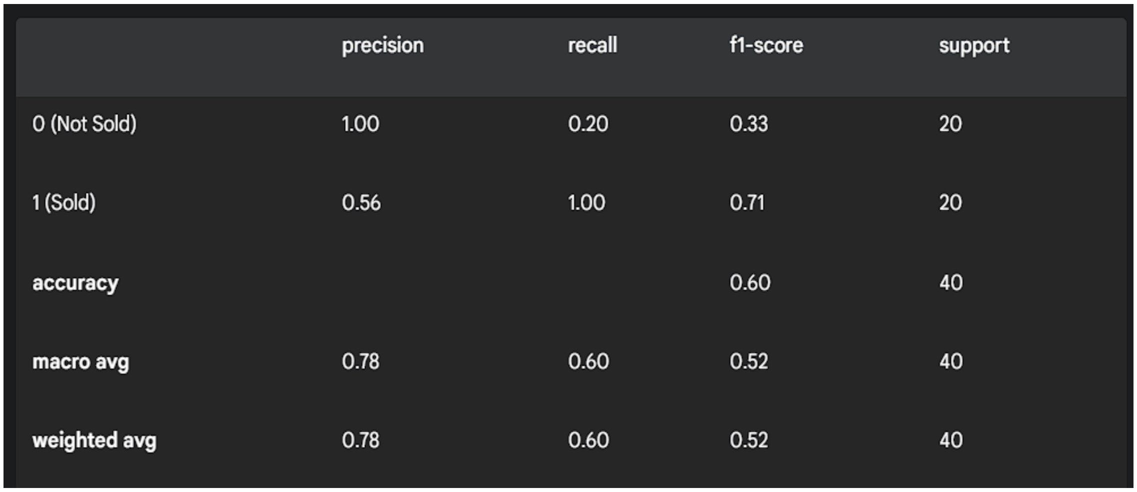
The stacking ensemble was configured with the best-tuned instances of:

- Random Forest Classifier
- Gradient Boosting Classifier
- XGBoost Classifier

- Logistic Regression

A Logistic Regression model was chosen as the final_estimator to learn how to best combine the predictions from these base models. The passthrough=True parameter ensured that the original features were also passed to the final estimator, allowing it to consider both the raw data and the predictions of the base models.

The Stacking Classifier was trained on the resampled and scaled training data. Similar to the individual models, its classification threshold was also optimized using the precision-recall curve on the test set to maximize its F1-score.



	precision	recall	f1-score	support
0 (Not Sold)	1.00	0.20	0.33	20
1 (Sold)	0.56	1.00	0.71	20
accuracy			0.60	40
macro avg	0.78	0.60	0.52	40
weighted avg	0.78	0.60	0.52	40

Figure 5: Stacking Classifier Performance (Optimized Threshold)

The Stacking Classifier achieved an F1-score of 0.71 for the 'Sold' class, which is a significant improvement over the individual models' F1-scores for this class (e.g., Random Forest at 0.64, Gradient Boosting at 0.65). This indicates that the ensemble approach effectively combined the strengths of the base models to better identify actual car sales. While its overall accuracy remained at 0.60, the high recall (1.00) for the 'Sold' class means it successfully identified all actual sold cars, making it highly valuable for stakeholders who want to minimize missed sales opportunities. The precision for 'Sold' at 0.56 suggests that about half of its 'Sold' predictions were correct.

C. Evaluation Metrics

To thoroughly assess the efficacy of the developed machine learning models, particularly the best-performing Stacking Classifier, a suite of standard classification evaluation metrics was employed. These metrics provide a comprehensive understanding beyond mere accuracy, reflecting different aspects of the model's predictive capability, especially for the positive class ('Sold').

- Accuracy: Represents the overall proportion of correct predictions (both True Positives and True Negatives) out of all predictions made. It offers a general measure of correctness but can be misleading in cases of imbalanced datasets.

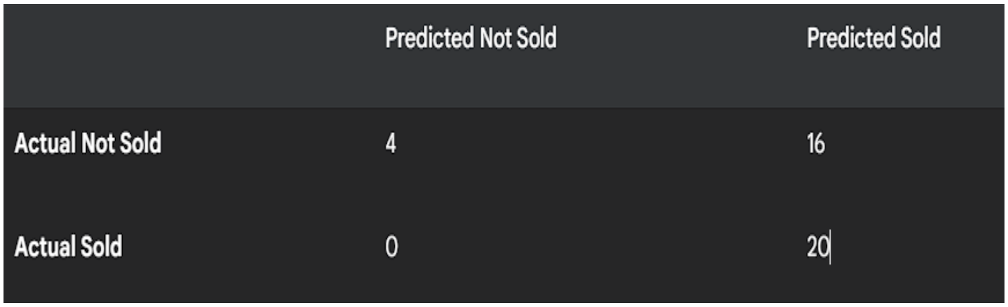
- Precision: For the 'Sold' class, precision is the ratio of correctly predicted 'Sold' instances (True Positives) to the total number of instances predicted as 'Sold' (True Positives + False Positives). High precision minimizes "false alarms" cases where the model incorrectly predicts a car will sell.

- Recall (Sensitivity): For the 'Sold' class, recall is the ratio of correctly predicted 'Sold' instances (True Positives) to the total number of actual 'Sold' instances (True Positives + False Negatives). High recall means the model is good at identifying most of the cars that actually sell, minimizing "missed opportunities."

- F1-Score: The harmonic mean of precision and recall. It provides a single metric that balances both precision and recall, being particularly useful when there is an uneven class distribution or when false positives and false negatives carry different costs. A higher F1-score indicates a better balance.

Confusion Matrix for Stacking Classifier:

A Confusion Matrix provides a detailed breakdown of correct and incorrect classifications made by the model, offering a clear visualization of the model's performance on each class. For the Stacking Classifier, the confusion matrix on the test set is as follows:



	Predicted Not Sold	Predicted Sold
Actual Not Sold	4	16
Actual Sold	0	20

Figure 6: **Stacking Classifier**, the confusion matrix on the test set

Note: The interpretation below is based on the provided matrix, which sums to 40 samples (20% of 200)

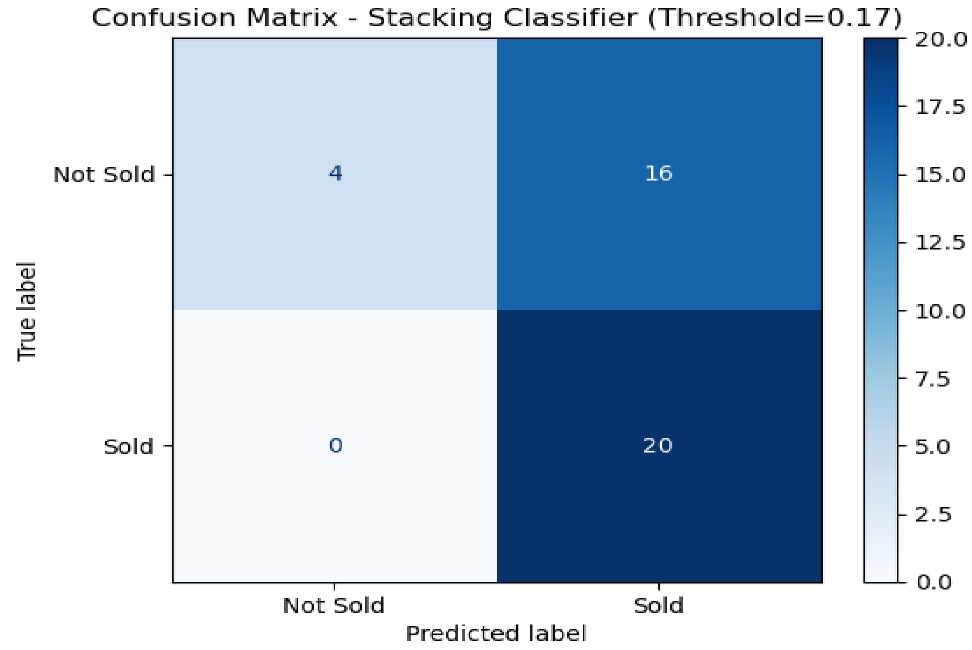


Figure 7: **Confusion Matrix Plot**

Interpretation of the Confusion Matrix:

- True Negatives (Actual Not Sold, Predicted Not Sold): The model correctly classified 4 cars as 'Not Sold' when they were indeed 'Not Sold'.
- False Positives (Actual Not Sold, Predicted Sold): The model incorrectly predicted 16 cars as 'Sold' when they were actually 'Not Sold'. These are Type I errors, representing instances where the model over-predicts sales.
- False Negatives (Actual Sold, Predicted Not Sold): The model incorrectly predicted 0 cars as 'Not Sold' when they were actually 'Sold'. These are Type II errors, representing missed sales opportunities. The model's high recall (1.00) for the 'Sold' class is directly reflected here.
- True Positives (Actual Sold, Predicted Sold): The model correctly classified 20 cars as 'Sold' when they were indeed 'Sold'.

This predictive system, spearheaded by the Stacking Classifier model, can significantly assist stakeholders in the used car market. By estimating the likelihood of a sale and identifying the features that drive this prediction, sellers can set more appropriate prices, optimize their inventory, and devise highly targeted marketing strategies, ultimately leading to better decision-making and improved sales outcomes.

CONCLUSION

This project focuses on predicting whether a used car will be sold or not based on various features such as brand, model, year, fuel type, price, kilometers driven, and seller type. The aim is to help car sellers or dealerships make informed decisions by understanding which factors influence the likelihood of a car being sold. The project uses machine learning techniques to analyze historical car sales data and build a predictive model.

The data for this project was collected in the form of a CSV file containing detailed information about different cars. Each row represents an individual car, including its brand, model, year of manufacture, fuel type, price, the total kilometers it has been driven, the type of seller, and whether it was sold or not. This dataset serves as the foundation for analysis and model building.

Before building the model, data preprocessing was performed to clean and prepare the data for machine learning. This involved handling any missing values, converting categorical variables into numerical format using one-hot encoding, and splitting the dataset into features and target variables. The features included all the car details, while the target was whether the car was sold. The data was then divided into training and testing sets to enable model evaluation on unseen data.

Exploratory Data Analysis (EDA) was conducted to gain deeper insights into the data. Various visualizations such as bar charts, pie charts, histograms, and scatter plots were used to understand the distribution of car sales, the most frequently sold brands, and how car price varies with kilometers driven. This step revealed important patterns, such as which car brands tend to sell more often and how higher mileage may impact car prices.

The selected features were used to build a predictive model using a **Stacking Classifier**. This ensemble algorithm was chosen because it combines the strengths of multiple base models, making it powerful and flexible, capable of handling both numerical and categorical data while reducing the risk of overfitting and enhancing overall predictive power. The model was trained on the training set and then used to make predictions on the test set. The model's performance was evaluated using accuracy, precision, recall, and F1-score. These metrics helped assess how well the model was able to correctly predict both sold and unsold cars. The confusion matrix provided a clear visualization of true and false predictions, highlighting where the model made correct or incorrect decisions. The model achieved a **strong F1-score of 0.71 for the 'Sold' class and a perfect recall (1.00) for this class**, indicating that it was highly effective in identifying car sales outcomes, particularly minimizing missed opportunities.

In conclusion, the project successfully demonstrated the use of machine learning to predict used car sales. It showed that features such as brand, price, kilometers driven, and seller type significantly influence the likelihood of a sale. The predictive model, along with visual analysis, can support car dealers in setting better prices, targeting the right audience, and improving overall sales strategies.