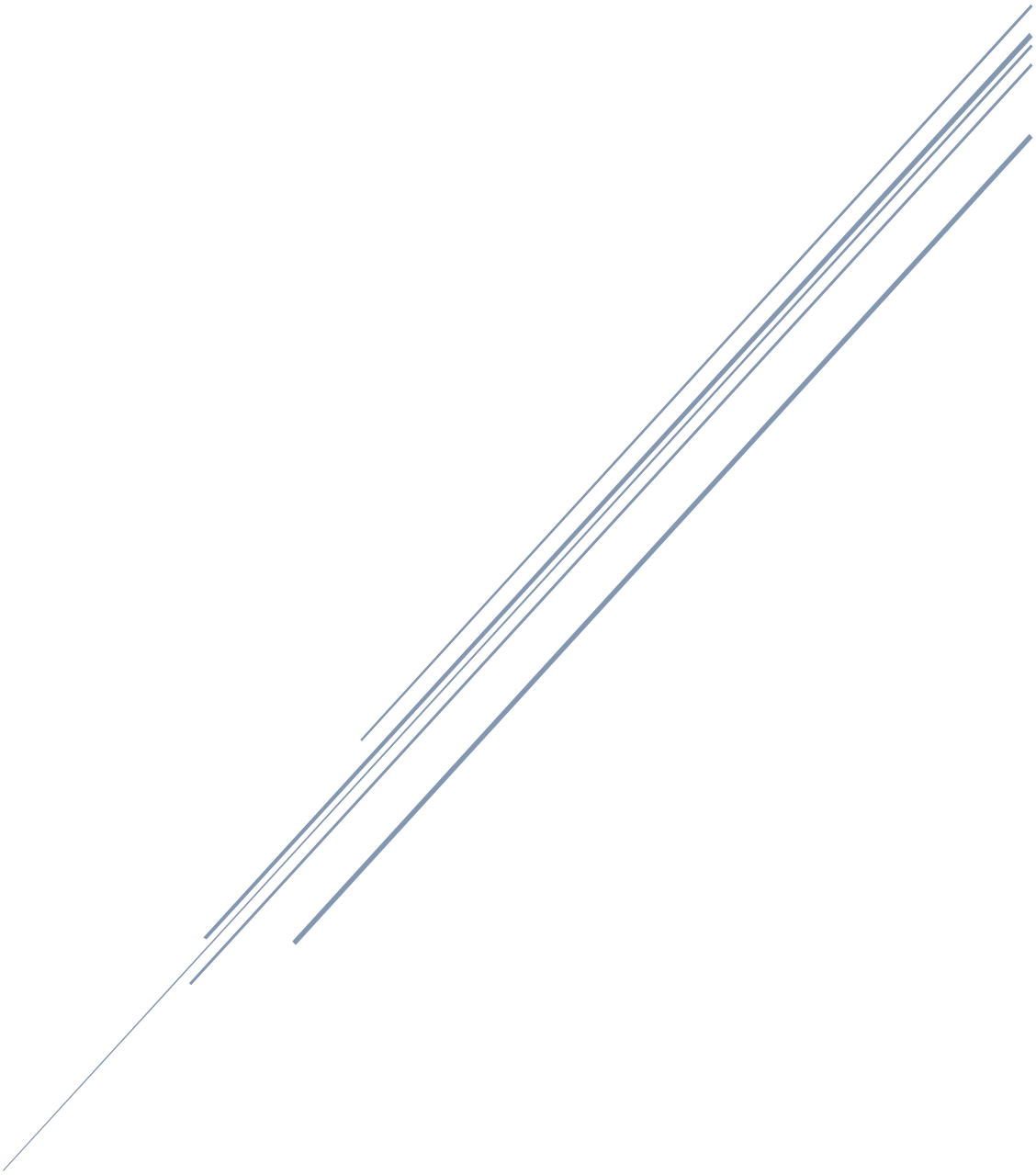
University of East London

ADVANCE DECISION MAKING

PREDICTIVE ANALYSIS & MACHINE LEARNING



University of East London
DS7003

Table of Contents

Abstract:

This project explores the use of machine learning algorithms to predict car acceptability based on categorical input features such as buying price, maintenance cost, number of doors, seating capacity, luggage boot size, and safety rating. Utilizing the Car Evaluation dataset from the UCI Machine Learning Repository, the study implements and compares three classification models: Logistic Regression, K-Nearest Neighbours (KNN), and Decision Tree to understand how these features influence consumer assessments of vehicles.

The evaluation results indicate that Logistic Regression provided the most stable and accurate performance, achieving the highest mean accuracy (93.2%) and Cohen's Kappa (0.853) through 10-fold cross-validation. This reflects its strong generalization ability and balanced classification across all target classes. KNN, optimized with a k value of 3, also performed well, with a mean accuracy of 86.1% and a Kappa of 0.675, though it showed greater sensitivity to data variation, particularly in minority class predictions. The Decision Tree model, while offering high interpretability and the best single-split accuracy (94.0%), demonstrated the lowest cross-validated performance (mean accuracy 78.2%, Kappa 0.534), indicating overfitting and limited generalizability. Overall, Logistic Regression emerged as the most dependable model for real-world decision support applications, with KNN as a viable alternative under optimal tuning, and Decision Trees requiring further refinement or ensemble methods to improve practical performance.

1. Introduction:

The Car Evaluation dataset, accessible via UCI, is a well-known standard for machine learning classification tasks. It is based on a hierarchical decision model introduced by Bohanec & Rajkovič in 1987 and replicates the process of assessing cars using a set of defined attributes. The dataset contains 1,728 records, each characterized by six categorical features: buying price, maintenance cost, number of doors, passenger capacity, luggage boot size, and safety rating. The target label, car acceptability, is categorized into four classes: unacceptable, acceptable, good, and very good.

This dataset is particularly suitable for educational and experimental purposes due to its:

- Well-structured format with no missing values,
- Categorical simplicity, and

- Real-world relevance in decision-making scenarios.

Although the dataset is synthetically generated, it closely reflects real-world car evaluation standards, making it an excellent tool for experimenting with classification algorithms like Decision Trees, Logistic Regression, and k-Nearest Neighbours (K-NN). It also offers a practical setting for tackling typical machine learning challenges such as class imbalance and encoding categorical variables.

By engaging with this dataset, learners and researchers can gain valuable insights into how different car attributes influence consumer decisions and develop predictive models that generalize well to unseen data.

2. Objectives:

The main goal of the Car Evaluation dataset is to support the creation and assessment of classification models that can determine a car's acceptability level, categorized as unacceptable, acceptable, good, or very good based on a range of categorical features. These features include factors such as buying cost, maintenance expense, number of doors, seating capacity, boot size, and safety level.

This data set serves several key purposes:

- **Educational Utility:** It provides a structured and accessible platform for students and practitioners to learn and apply machine learning techniques, particularly classification algorithms.
- **Model Development:** It enables the testing and comparison of various machine learning models, such as Decision Trees, Logistic Regression, and k-Nearest Neighbours, in a controlled environment.
- **Feature Impact Analysis:** It allows for the exploration of how different car attributes influence consumer acceptability, offering insights into decision-making processes.
- **Handling Categorical and Imbalanced Data:** The dataset presents challenges such as categorical encoding and class imbalance, making it a valuable resource for practicing data preprocessing and model tuning strategies.

Overall, the dataset is designed to support both foundational learning and advanced experimentation in supervised machine learning, particularly in the context of real-world decision support systems.

3. Reason for Using the Car Evaluation Dataset:

This dataset is selected for this study due to its unique combination of simplicity, practical relevance, and educational value. Originally derived from a decision-making model, the dataset simulates real-world scenarios in which consumers evaluate vehicles based on key attributes such as price, maintenance cost, safety, and capacity. These features are not only intuitive but also critical in real-life car purchasing decisions, making the dataset highly relatable and applicable.

Several factors justify the use of this dataset:

- **Beginner-Friendly Structure:** With only six input features and a clearly defined target variable, the dataset is ideal for those new to machine learning. It allows learners to focus on model development and evaluation without being overwhelmed by data complexity.
- **Categorical Nature:** All features are categorical, providing an excellent opportunity to practice encoding techniques and explore how machine learning models handle non-numeric data.
- **No Missing Values:** The dataset is clean and complete, eliminating the need for extensive preprocessing and allowing for a more focused exploration of classification algorithms.
- **Class Imbalance Challenge:** The dataset presents a realistic challenge in the form of class imbalance, where most instances are labeled as "unacceptable." This encourages the application of advanced techniques such as oversampling, undersampling, and class weighting.
- **Versatility for Model Comparison:** It supports the implementation and comparison of a wide range of classification algorithms, including Decision Trees, Logistic Regression, k-Nearest Neighbors, and Random Forests, among others.
- **Widely Recognized Benchmark:** As a standard dataset in the UCI repository, it has been extensively used in academic research and machine learning education, allowing for meaningful comparisons with existing studies.

In summary, this dataset offers a balanced mix of accessibility In and analytical depth, making it an ideal choice for both foundational learning and advanced experimentation in supervised machine learning.

4. Literature Review:

The Car Evaluation dataset is a popular choice for exploring and practicing machine learning techniques, particularly for classification problems. Decision Trees are widely appreciated for their clarity and straightforward implementation. They enable users to trace decision-making processes and see how attributes such as safety and purchase price impact the classification of car acceptability. Research indicates that Decision Trees yield strong performance on this dataset, particularly when pruning methods are applied to reduce overfitting (Quinlan, J.1986).

These machine learning methods are widely recognized for their transparency and strong predictive capabilities. Decision Trees are valued for their straightforward interpretability, while Random Forests enhance predictive accuracy and reduce the risk of overfitting by aggregating the outputs of multiple decision trees. In a recent study on used car price prediction, Random Forests achieved a notably high R^2 score of 0.77, outperforming individual Decision Trees. This makes them a preferred approach for regression tasks involving structured datasets (Springer, 2023).

K-Nearest Neighbors (K-NN) is commonly employed as a baseline model in classification tasks. It delivers decent results when the data is well-preprocessed and normalized. However, its effectiveness can diminish in the presence of high-dimensional categorical features unless these are appropriately encoded (Cover, T. 1967).

Logistic Regression is frequently used to estimate the probability of a car being classified as 'unacceptable' based on features such as buying cost, maintenance expense, and safety level. It provides interpretable coefficients and is grounded in solid statistical theory. However, its performance may be limited when modeling complex, non-linear relationships, where tree-based methods often perform better (Hosmer, 2013).

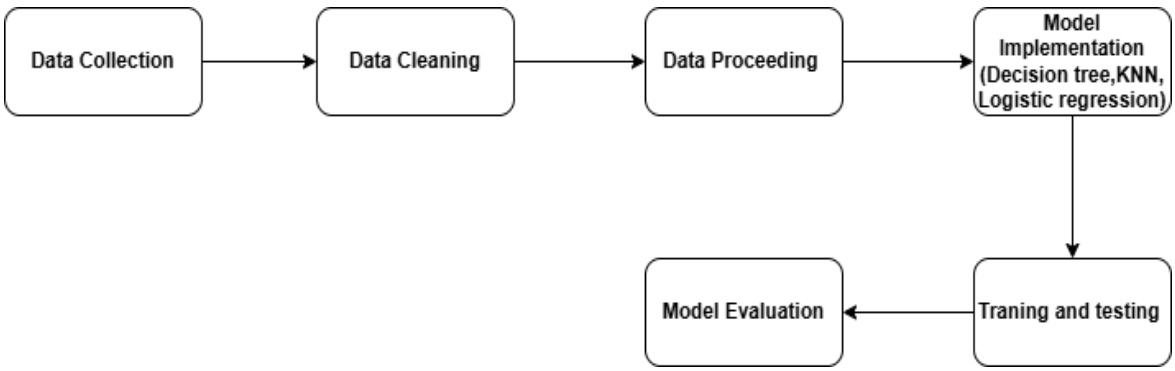
Random Forests tend to outperform simpler models thanks to their ensemble approach, which captures complex feature interactions. They are also more resistant to overfitting and handle class imbalance effectively (Breiman, L. 2001).

SVMs, especially with Radial Basis Function (RBF) kernels, have been used in ensemble models to improve classification accuracy. They are effective in high-dimensional spaces and can handle non-linear boundaries well (Govindarajan, M, 2014).

Naive Bayes classifiers are fast and efficient, particularly for categorical data. While they assume feature independence, they still perform competitively on this dataset and are often used as a baseline (McCallum, A. 1998).

Neural Networks have been utilized on this dataset to model intricate, non-linear patterns. With proper tuning, they can achieve performance comparable to or even better than traditional models, although they typically demand greater computational power (Goodfellow, I. 2016).

5. Flowchart:



6. Methodology

I. Data Collection:

This dataset was designed to replicate how vehicles are assessed based on attributes such as price, safety, and seating capacity. Although the exact methodology behind its creation isn't fully detailed, the dataset was intentionally constructed to represent realistic evaluation scenarios. It includes a diverse range of attribute combinations paired with corresponding acceptability ratings, making it highly suitable for machine learning applications. Hosted on the UCI Machine Learning Repository, a reputable and widely used source of open-access datasets, it is easily accessible for students, researchers, and developers. This makes it an excellent starting point for learning how different car features impact overall acceptability and building classification models in a beginner-friendly environment.

II. Explanatory Data sets:

In this dataset, the variables are divided into two main types: independent variables (features) and the target variable.

i. Target Variable (Dependent Variable):

class: This is the variable the model aims to predict. It represents the acceptability of a car, categorized into:

- **unacc** (unacceptable)
- **acc** (acceptable)
- **Good**
- **vgood** (very good)

ii. Independent Variables (Features)

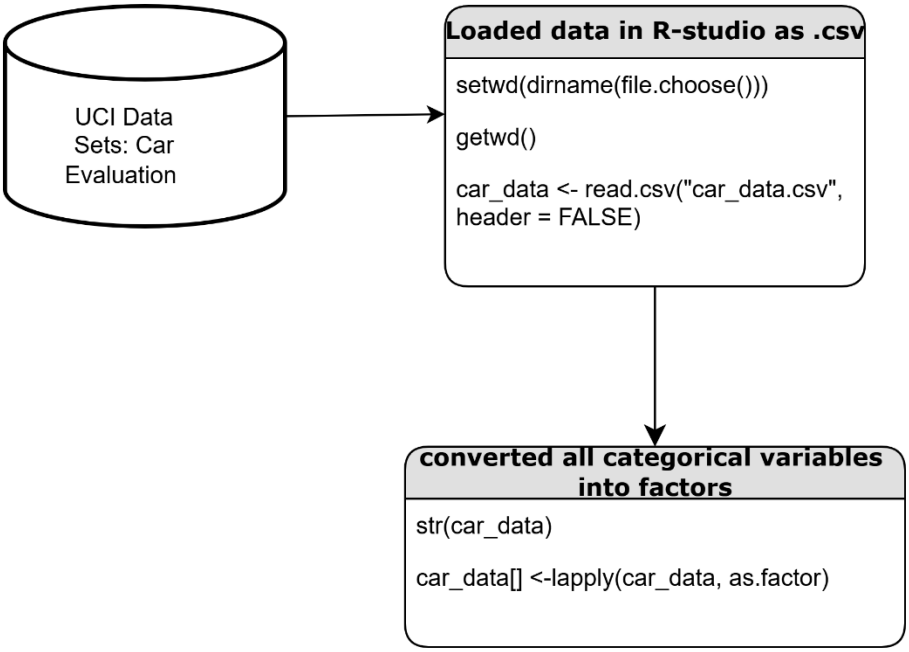
These are the input features used to predict the target variable:

- Buying: Buying price of the car
  - Values: vhigh, high, med, low
- Maint: Maintenance cost
  - Values: vhigh, high, med, low
- Doors: Number of doors
  - Values: 2, 3, 4, 5 more
- Persons: Passenger capacity
  - Values: 2, 4, more
- lug_boot: Size of luggage boot
  - Values: small, med, big

- Safety: Safety rating
  - Values: low, med, high

III. Data cleaning & Preprocessing:

A. Converting Categorical variables into factorial variables:



In this dataset, all seven variables, including features such as price, safety, and seating capacity, are categorical. The target variable, car acceptability, is also categorical with classes like "low," "med," and "high." Since machine learning models typically require numerical input, working directly with raw categorical data can lead to inaccurate results. To address this, the categorical variables were converted into factor (factorial) form using R functions, allowing the data to be properly processed and analysed.

- **To fix it:**

The line of code **car_data[] <- lapply**(car_data, as.factor) uses base R functions to convert all columns in the **car_data** dataframe into factors (categorical variables). Here's what each component does:

- **lapply():**

A base R function that applies a specified function to each element of a list in this case, each column of the dataframe.

- **as.factor():**

  Converts a vector (or column) into a factor, which is R's data type for categorical variables.

- **car_data[] <-:**

The empty square brackets **[]** ensure that the structure of the dataframe is preserved during assignment. Without them, the result would be a list instead of a dataframe.

B. For check Missing Values:

```
> # Check for missing values
> sum(is.na(car_data))
[1] 0
```

The dataset has no missing values

C. Imbalanced Classes:

- The Issue: The dataset is unbalanced. Most cars are labeled as "**unacceptable**" (70%), while very few are labeled as "**good**" or "**very good**" (only 4% each).
- **To fix it:**
  - Oversampling: Add more examples of the rarer classes (like "good" and "very good") to balance things out.
  - Underdamping: Reduce the number of examples in the overrepresented class ("unacceptable").
  - Class Weighting: Adjust your model to give more importance to the rarer classes during training.

III. Summary of the Datasets

```
> summary(car_data)
   buying       maint        doors      persons      lug_boot    safety       class
 high :432   high :432   2    :432   2    :576   big  :576   high:576   acc  : 384
 low  :432   low  :432   3    :432   4    :576   med  :576   low :576   good :  69
 med  :432   med  :432   4    :432   more:576   small:576   med :576   unacc:1210
 vhigh:432   vhigh:432   5more:432                                      vgood:  65
```

The car evaluation dataset offers a detailed snapshot of the distribution of categorical variables across six input features and one target class. The dataset contains 1,728 instances, as reflected by the total counts across each attribute. The buying and maint attributes (representing purchase price and maintenance cost) each include four categories vhigh, high, med, and low with an equal distribution of 432 instances per category, indicating a well-balanced representation. Similarly, the doors attribute (number of doors) is evenly split among the values 2, 3, 4, and 5more, each appearing 432 times. The persons variable (seating capacity) is uniformly distributed across 2, 4, and more, with 576 instances in each category. The lug_boot (luggage boot size) and safety (safety rating) attributes also show balanced distributions across their three respective categories (small, med, big and low, med, high), each with 576 occurrences.

In contrast, the target variable (class) is highly imbalanced. The unacc (unacceptable) class dominates with 1,210 instances, while acc (acceptable), good, and vgood (very good) are significantly underrepresented, with 384, 69, and 65 instances, respectively. This pronounced

class imbalance is a critical factor in model development and evaluation, as it can bias classifiers toward the majority class and hinder accurate prediction of minority classes.

IV. Distribution of the data sets:

Figure 1 illustrates that all categorical variables are well-balanced and evenly distributed after encoding; there is no need for further normalization or scaling before applying classification algorithms.

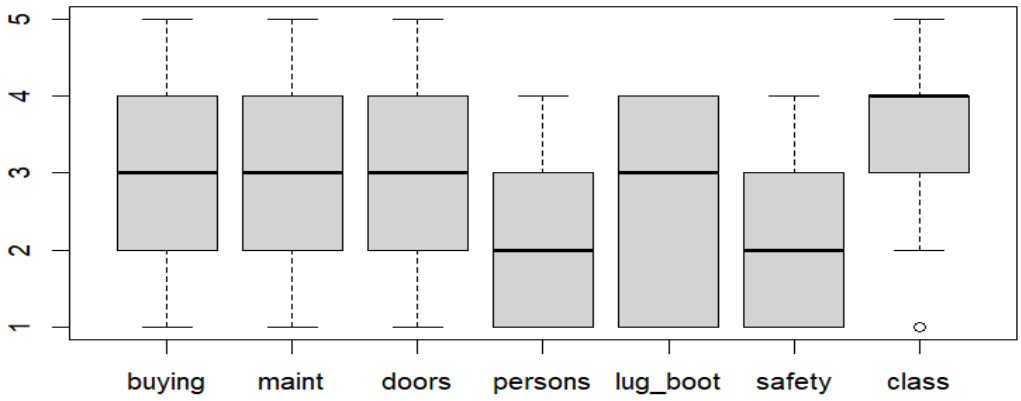


Fig1: Box plot

V. Measure of Association: Cramer's V – association of the target variable with the independent variable:

Since all variables in this dataset are categorical, traditional correlation metrics aren't suitable. Instead, we use association measures tailored for categorical data. One such measure is Cramér's V, which quantifies the strength of association between two categorical variables. It ranges from 0 (no association) to 1 (perfect association), making it a useful tool for evaluating the relationship between the target variable (class) and each of the input features.

The interpretation of Cramér's V depends on the degrees of freedom (df), which is calculated as: **df=min (number of rows−1, number of columns−1)**

The strength of association can be interpreted using the following thresholds based on the degrees of freedom (Bobbitt, Z., 2021).

a. Result and data visualization:

Cramér's V is used to assess the strength of association between the target variable (class, indicating car acceptability) and each of the independent features. The values of Cramér's V range from 0 (no relationship) to 1 (perfect relationship), providing insight into how strongly each feature is linked to the classification outcome.

i. cramers_v results interpretation:

```
cramers_v_results
    buying      maint      doors    persons   lug_boot     saf
ety
0.19106337 0.16605242 0.04475753 0.32779105 0.12416623 0.37241
487
```

Interpretation of Cramér's V Values:

Variable	Cramér's V	Cramér's Strength	Interpretation
safety	**0.37**	Moderate	Safety has the strongest influence on car acc eptability. This suggests that higher safety ra tings are more likely to be associated with b -etter class labels.
persons	**0.33**	Moderate	Passenger capacity is also a significant factor in determining car acceptability.
Buying	**0.19**	Weak	Buying prices have a weak but noticeable inf -luence.
Maint	**0.17**	Weak	Maintenance cost has a similar weak associa -tion.
lug_boot	**0.12**	Very Weak	Luggage boot size has minimal impact.
doors	**0.04**	Negligible	The number of doors has almost no influenc e on the car's acceptability (target variable).

Table 1: Interpretation of Cramér's V Values

b. Plot of Cramer's V – association of the target variable with the independent variable:

Fig 2: Plot of Camer's V with target variable

Figure 2: Cramer's V plot illustrates the strength of association between each independent variable and the target variable (class). Cramer's V values range from 0 (no association) to 1 (perfect association), helping to identify which features are most influential in predicting the target.

Summary of association:

- **Safety (0.37)** and **Persons (0.33)** show the strongest associations with the target variable, indicating they are the most informative features for classification.
- **Buying (0.19)** and **Maint (0.17)** have moderate associations, suggesting they contribute to the model but are less influential.
- **Lug_boot (0.12**) shows a weaker relationship.
- **Doors (0.04)** have the lowest association, implying minimal impact on the target prediction.

VI. Data Splitting:

To thoroughly assess the models' predictive performance and ability to generalize, the dataset was partitioned into two distinct subsets: 70% allocated for training and 30% reserved for testing. The training subset was utilized to build and calibrate the models by identifying patterns within the data, whereas the testing subset functioned as an independent dataset to evaluate the models' effectiveness on unseen data. This strategy helps reduce the risk of overfitting and offers a more realistic estimate of how the models would perform when applied to real-world situations.

7. Machine learning techniques:

The dataset models a real-world decision-making scenario: assessing cars based on attributes such as price, maintenance cost, safety, and seating capacity. ML models can automate this evaluation process, offering data-driven predictions in place of manual or rule-based systems. Since all features in the dataset are categorical, it is particularly well-suited for machine learning algorithms that handle nominal data effectively such as Decision Trees, KNN, Logistic regression. These techniques not only enable accurate classification but also help identify which features have the greatest impact on determining car acceptability.

I. Logistic Regression:

Logistic regression is a supervised learning technique used to predict the likelihood of a categorical outcome based on one or more input variables (Hosmer, Lemeshow & Sturdivant, 2013). Unlike linear regression, which is suited for continuous outcomes, logistic regression is tailored for classification problems where the target variable is either binary or has multiple categories. It employs the logistic (sigmoid) function to ensure that predicted values remain between 0 and 1. When applied to the Car Evaluation dataset

(Bache & Lichman, 2013), logistic regression offers a straightforward and easy-to-understand approach for estimating a car's acceptability based on factors such as buying price, maintenance cost, and safety levels.

Given that the target variable (class) comprises four categories: unacceptable (unacc), acceptable (acc), good, and very good (vgood), a multinomial logistic regression model (also known as softmax regression) is appropriate (Agresti, 2018).

a. Confusion Matrix and Statistics

Prediction	Reference acc	good	unacc	vgood
acc	95	2	15	3
good	3	16	0	0
unacc	15	0	348	0
vgood	2	2	0	16

The confusion matrix, along with related classification metrics, offers a detailed assessment of how well the logistic regression model performs on a multiclass classification problem. In the matrix, each cell located at position (i, j) represents the number of times instances belonging to the true class j were incorrectly or correctly classified as class i by the model.

- Diagonal elements (95, 16, 348, 16) represent correct classifications (true positives per class).
- Off-diagonal elements represent misclassifications.
- The majority class, unacc, is predicted with the highest accuracy, achieving 348 correct classifications. In contrast, the minority classes, good and vgood, have fewer instances and demonstrate only moderate predictive performance.

Class-wise Performance Analysis:

- The model shows strong performance in identifying the "unacc" (unacceptable) class, with true positive (TP) 348 instances correctly classified. This suggests a high level of accuracy in detecting this predominant category.
- The "acc" (acceptable) class shows a strong correct prediction count of true positive (TP) 95 instances. However, the model also misclassified false negatives (FN) 20 instances as "unacc" and 3 as "vgood", indicating some overlap or confusion between these categories.
- For the less representative classes, good has a low number of true positive (TP) predictions (16), with 4 false negatives (FN), indicating that all actual good instances were correctly identified. Similarly, vgood has 16 true positives (TP). Despite the limited sample sizes, the model performs reasonably well. However, there is some confusion between these two classes, as 4 vgood false negative (FN) instances were

incorrectly classified as good, highlighting a challenge in distinguishing between these closely related categories.

b. Heat map of the logistic regression Confusion matrix:

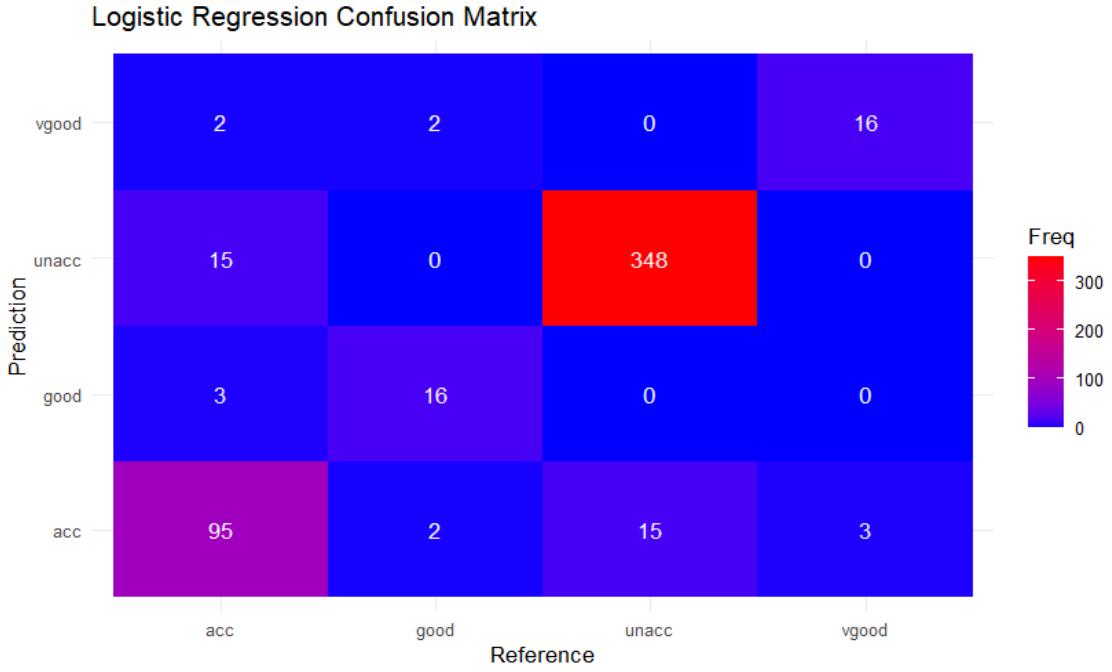


Figure 3: Heat map of the logistic regression Confusion matrix

Figure 3 heatmap illustrates the confusion matrix generated by a logistic regression model applied to the Car Evaluation Dataset. It displays predicted categories along the rows and actual categories along the columns, with the intensity of the color reflecting how frequently each prediction occurred.

Confusion Matrix Heatmap Structure:

- **x-axis (Reference):** represents the actual class labels.
- **y-axis (Prediction):** represents the predicted class labels.
- **Diagonal cells** (unacc–unacc, good–good) show correct predictions.
- **Off-diagonal cells** (acc–unacc, vgood–acc) show misclassifications.
- **Cell Values:** Count of observations for each true-predicted class pair
- **Color Scale (Freq):** Red represents the highest frequencies (above 300 correct predictions for (unacc), purple indicates relatively high values (around 95), light blue shows moderate frequencies, and blue corresponds to the lowest values (rare misclassifications).

c. Overall Statistics

```
                Accuracy : 0.9188
                  95% CI : (0.8918, 0.9408)
    No Information Rate : 0.7021
    P-Value [Acc > NIR] : < 2.2e-16

                   Kappa : 0.8213
```

```
Mcnemar's Test P-Value : NA
```

- **Accuracy (91.88%):** The logistic regression model's overall performance on this Dataset offers a clear picture of its classification capabilities. With an accuracy of 91.88%, the model correctly identified nearly 92 out of every 100 cases across the four categories: acceptable (acc), good, unacceptable(unacc), and very good(vgood). This high level of accuracy reflects the model's strong alignment with the underlying patterns in the data.
- **Confidence Interval:** The 95% confidence interval (CI) for the model's accuracy, spanning from 89.18% to 94.08%, suggests that there is a 95% likelihood that the true accuracy of the model, when applied to the broader population, lies within this range. The relatively tight interval reflects stable model performance with minimal variability, strengthening confidence in both the reliability and generalizability of the model's predictions (James G., 2021).
- **No Information Rate (NIR):** The No Information Rate (NIR) of 70.21% reflects the accuracy that would result from always predicting the most frequent class, most likely the unacceptable category, which dominates the dataset.
- **P-Value [Acc > NIR]:** P-Value [Acc > NIR]: The p-value for the comparison between the model's accuracy and the No Information Rate (NIR) is less than 2.2e-16, offering compelling evidence to reject the null hypothesis that the model performs no better than simply predicting the most common class. This exceptionally small p-value demonstrates that the logistic regression model achieves a statistically significant improvement over a baseline or naive classifier (Hosmer, Lemeshow & Sturdivant, 2013).
- **Kappa (0.8213):** Cohen's Kappa, with a value of 0.8213, quantifies the level of agreement between the model's predictions and the actual classifications, adjusting for the possibility of chance agreement. Based on standard interpretation guidelines, a Kappa score above 0.80 is considered to reflect "almost perfect" agreement (Landis & Koch, 1977). This indicates that the model's predictions are highly consistent and unlikely to be due to random chance.
- **McNemar's Test:** The McNemar's Test p-value is marked as not applicable (NA), which is appropriate in this case. McNemar's test is generally used to evaluate differences in paired proportions, typically within binary classification tasks. Since this analysis involves a multi-class classification problem without directly paired binary outcomes, the test is not suitable here (Kuhn & Johnson, 2019)

d. Statistics by Class:

```
                     Class: acc Class: good
```

Sensitivity	0.8261	0.80000
Specificity	0.9502	0.99396
Pos Pred Value	0.8261	0.84211
Neg Pred Value	0.9502	0.99197
Prevalence	0.2224	0.03868
Detection Rate	0.1838	0.03095
Detection Prevalence	0.2224	0.03675
Balanced Accuracy	0.8882	0.89698
	Class: unacc	Class: vgood
Sensitivity	0.9587	0.84211
Specificity	0.9026	0.99197
Pos Pred Value	0.9587	0.80000
Neg Pred Value	0.9026	0.99396
Prevalence	0.7021	0.03675
Detection Rate	0.6731	0.03095
Detection Prevalence	0.7021	0.03868
Balanced Accuracy	0.9306	0.91704

The class-specific metrics for the logistic regression model provide a detailed and nuanced view of its ability to distinguish between the four target categories: acceptable (acc), good, unacceptable (unacc), and very good (vgood) within this Dataset. These metrics go beyond overall accuracy, offering deeper insights into the model's performance across individual classes. They are especially valuable for evaluating how well the model handles class imbalance and navigates complex classification boundaries.

Class-wise Interpretation:

- **Class: Acceptable (acc):**
  - **Sensitivity:** For the acceptable class, the model achieved a sensitivity (or recall) of 0.8261, meaning it correctly identified 82.61% of all actual instances labeled as "acc.
  - **Specificity:** A specificity of 0.9502 means that 95.02% of cases that did not belong to the "acceptable" class were accurately identified and excluded by the model. This high level of specificity, coupled with strong recall, underscores the model's effectiveness in making precise and reliable distinctions in its classifications.
  - **Positive & Negative predictive value (PPV) & (NPV) :** The **PPV** of 0.8261 and the **NPV** of 0.9502 further affirm the model's robustness in classifying instances within the "acceptable" category. These metrics demonstrate that the model's predictions were consistently trustworthy, both when confirming membership in the class and when accurately excluding it.
    - **Balanced Accuracy: Class: Good:** The balanced accuracy of 0.8882, computed as the average of sensitivity and specificity, indicates that the logistic regression model delivers stable and reliable classification performance across all classes, regardless of class imbalance. This metric is particularly informative given the moderate prevalence

of the "acceptable" class at 22.24%, ensuring that both minority and majority classes are evaluated fairly.

- **Sensitivity:** Although the good class has a relatively low prevalence of just 3.87%, the model achieves a sensitivity of 0.8000, indicating that 80% of actual instances in this category were correctly identified.
- **Specificity:** The specificity for the good class is notably high at 0.99396, indicating that 99.40% of instances not belonging to this class were correctly identified as such by the model.
- **Positive & Negative predictive value (PPV) & (NPV):** With a PPV of 84.21%, the logistic regression model accurately identified good instances in most of its predictions. Additionally, the NPV of 99.20% indicates that the model was highly reliable in ruling out non-good cases.
- **Balanced Accuracy:** The balanced accuracy of 0.89698 is particularly notable, reflecting the model's strong performance even when dealing with underrepresented classes, a common challenge in multi-class imbalanced datasets (Sokolova & Lapalme, 2009).

- **Class: Unacceptable (unacc)**
  - **Sensitivity:** The unacceptable class, which is the most prevalent in the dataset (70.21%), is where logistic regression performs exceptionally well. It achieves a sensitivity of 95.87%, meaning the model correctly identifies nearly all instances of this dominant class.
  - **Specificity:** The model also achieves a specificity of 90.26% for the unacceptable class, meaning it correctly excludes non-unacc instances with high accuracy.
  - **Positive & Negative predictive value (PPV) & (NPV):** The model demonstrates strong predictive reliability for the unacceptable class, with a positive predictive value (PPV) of 95.87% and a negative predictive value (NPV) of 90.26%, confirming its effectiveness in both affirming and rejecting class membership for this dominant category.
  - **Balanced Accuracy:** The balanced accuracy of 0.9306 further highlights the model's consistent performance in classifying the unacceptable class, effectively capturing true positives while minimizing false positives.
- **Class: Very Good (vgood):**
  - **Sensitivity:** Despite vgood being a minority class with a prevalence of just 3.68%, the logistic regression model performs impressively, achieving a sensitivity of 0.8421.
  - **Specificity:** Despite vgood being a minority class with a prevalence of only 3.68%, the logistic regression model exhibits strong performance, achieving a sensitivity of 0.8421 and a high specificity of 0.99197.

- **Positive & Negative predictive value (PPV) & (NPV):** The PPV of 0.8000 indicates that 80% of the instances predicted as vgood were correct, while the NPV of 0.99396 highlights the model's excellent ability to accurately identify non-vgood cases.
- **Balanced Accuracy:** The balanced accuracy of 0.91704 further highlights the model's robustness and its capacity to generalize effectively when identifying instances of the vgood class, despite challenges posed by data sparsity. This strong performance is particularly significant, as minority classes in multinomial logistic regression models frequently suffer from reduced recall or precision due to their lower representation in the dataset (Hosmer, Lemeshow, & Sturdivant, 2013).

e. Summary of model:

```
> summary(logit_model)
Call:
multinom(formula = class ~ ., data = train_data)

Coefficients:
      (Intercept) buyinglow buyingmed buyingvhigh  maintlow  maintmed maintvhigh
G good    -122.2134 53.296578 48.401225  -15.442963 52.049681 47.046284 -46.233480
unacc    100.6527 -4.970708 -4.246474    1.911548 -3.108734 -3.701398   2.711399
vgood   -128.3989 93.240367 65.903978   -2.923801 37.519003 33.555243 -55.007811
         doors3    doors4 doors5more   persons4 personsmore lug_bootmed
good   1.507606  3.881414   4.008204   29.46429    29.69914   -2.386079
unacc -1.865846 -2.704240  -2.481219 -102.03695  -101.99163    1.257313
vgood 22.723606 28.120824  28.466709   33.83347    33.71598  -25.967147
      lug_bootsmall safetylow  safetymed
good      -9.730427 -29.27360  -8.085740
unacc      4.517917  91.18067   3.206985
vgood    -82.596709 -17.49995 -70.501570

Std. Errors:
      (Intercept)  buyinglow  buyingmed  buyingvhigh   maintlow   maintmed
good    0.9007675  0.9516353  0.4868701          NaN  1.0165935  0.5156886
unacc   0.3673418  0.6809363  0.6099738 4.554526e-01  0.5538754  0.6128267
vgood  21.3219280 31.9762595 53.2817371 6.381508e-07 85.2731950 85.2617551
        maintvhigh     doors3     doors4 doors5more   persons4 personsmore
good           NaN  1.2750947  1.4180831  1.3470098  0.5884541   0.5964856
unacc 4.917504e-01  0.4885375  0.5538829  0.5002767  0.2511704   0.2461444
vgood 8.649632e-06 85.2730528 85.2559537 85.2555045 10.6730603  10.6778671
      lug_bootmed lug_bootsmall    safetylow   safetymed
good    1.1268599   2.342409756          NaN   1.9692747
unacc   0.4562017   0.578038293 9.176028e-07   0.4476732
vgood  85.2545239   0.002691047 9.172489e-07 282.3629029

Residual Deviance: 300.2213
AIC: 396.2213
```

i. Model Overview :

The output reflects the results of a multinomial logistic regression model, estimated using the multinom() function. In this model, the dependent variable is the categorical outcome class,

which includes categories such as good, unacc, and vgood. The independent variables consist of various car attributes, including buying price, maintenance cost, number of doors, and safety. The model estimates the log-odds of each class category relative to a baseline or reference category in multinomial logistic regression in R (nnet package), the baseline is automatically the first level alphabetically.

ii. Coefficients Interpretation:

The coefficients in the multinomial logistic regression model represent the change in the log-odds of an observation being classified into a specific outcome category (good, unacc, or vgood) relative to the reference category (acc), for a one-unit increase in each predictor variable, assuming all other predictors remain constant.

- **Class = good:**
  - For the good class, the model yields large positive coefficients for the predictors buyinglow (53.30) and maintlow (52.05), suggesting that vehicles with low buying prices and low maintenance costs are significantly more likely to be classified as good rather than acceptable.
  - The positive coefficients for persons4 (29.46) and personsmore (29.70) indicate that vehicles with higher passenger capacity are strongly associated with the good class, suggesting that increased seating significantly boosts the likelihood of a car being classified as good rather than acceptable.
  - The negative coefficients for safetylow (–29.27) and lug_bootsmall (–9.73) suggest that vehicles with low safety ratings and small luggage capacity are significantly less likely to be classified as good compared to acceptable.
- **Class = unacc:**
  - The negative coefficients for variables such as buyinglow (–4.97) and maintlow (–3.11) indicate that vehicles with lower purchase and maintenance costs are less likely to be classified as unacceptable compared to acceptable. This suggests that affordability and low upkeep reduce the likelihood of a car being deemed unacceptable.
  - The strongly negative coefficients for persons4 (–102.04) and personsmore (–101.99) indicate that vehicles with higher passenger capacity are significantly less likely to be classified as unacceptable. This suggests that limited seating capacity greatly increases the likelihood of a car being deemed unacceptable, which aligns with practical expectations.
  - The extremely large positive coefficient for safetylow (91.18) suggests that vehicles with low safety ratings are overwhelmingly likely to be classified as unacceptable

rather than acceptable. This finding aligns well with practical expectations, as safety is a critical factor in vehicle evaluation.

- **Class = vgood:**
  - The high positive coefficients for buyinglow (93.24) and maintlow (37.52) indicate that vehicles with low purchase prices and low maintenance costs are significantly more likely to be classified as very good rather than acceptable. This reinforces the strong influence of affordability on top-tier classification.
  - The notably high coefficients for persons4 (33.83) and personsmore (33.72) further reinforce the idea that greater passenger capacity significantly enhances acceptability.
  - The substantial negative coefficients for safetymed (-70.50) and safetylow (-17.50) suggest that vehicles with medium or low safety ratings are less likely to be classified as vgood, highlighting consumers' strong preference for safer cars.

iii. Standard Errors & Significance:

- The standard errors measure the uncertainty around each coefficient estimate.
- The consistently low standard errors for the unacc class indicate that the estimates are highly reliable and precise.
- In the 'vgood' class, several standard errors are exceptionally large (around 85 for multiple predictors), indicating that the estimates may be unstable, possibly due to issues like data sparsity, perfect separation, or multicollinearity associated with this rare class.
- The presence of NaNs in certain standard errors (for 'buyingvhigh' and 'maintvhigh' in the 'good' class) suggests that some parameters could not be estimated, likely due to perfect prediction or lack of variability in those categories, possibly caused by collinearity.

iv. Model Fit Statistics:

- The Residual deviance is 300.22. While this value alone isn't inherently meaningful, a lower deviance generally indicates a better-fitting model. It becomes particularly useful when comparing nested models to assess relative fitness.
- The Akaike Information Criterion (AIC) is 396.22. This metric balances model fits with complexity, where lower AIC values suggest a more optimal model by penalizing unnecessary complexity (Burnham & Anderson, 2002).

v. Bar plot of Influence of predictor variables:

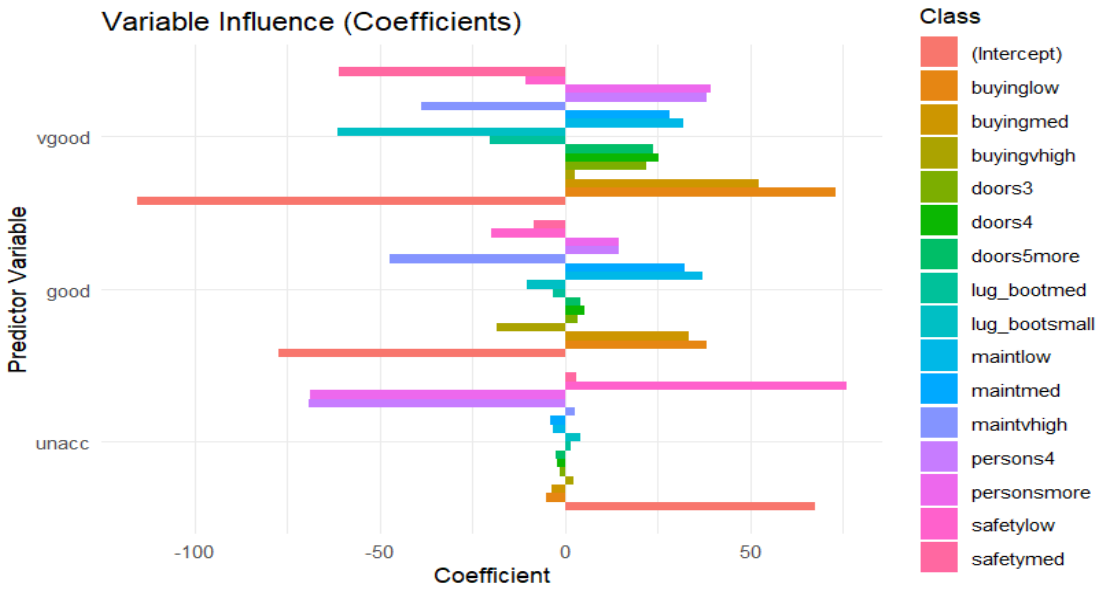


Fig 4 : Bar plot of the influence of predictor variables

Figure 4 illustrates the estimated coefficients from a multinomial logistic regression model used to predict car acceptability across four categories: unacc, acc, good, and vgood. It provides a clear overview of how various car features influence classification outcomes. Among the predictors, safety and passenger capacity stand out as the most influential, with safety showing particularly strong differentiation between the 'unacceptable' and 'very good' categories. In contrast, features like the number of doors and luggage boot size have minimal impact across all classes. These findings highlight the critical role of safety and practicality in shaping consumer perceptions of car quality.

a. **Axes Interpretation:**

- **X-axis (Coefficient Values):**
  - Represents the magnitude and direction of influence each predictor has on the model.
  - Positive coefficients indicate a positive association with the target class (i.e., increasing the predictor increases the likelihood of the class).
  - Negative coefficients indicate a negative association.
- **Y-axis (Predictor Variables):**

Lists of the categorical and ordinal features used in the model, such as:

  - Car acceptability levels (vgood, good, unacc)
  - Buying price (buyinglow, buyingmed, buyinghigh)
  - Number of doors (doors3, doors4more)
  - Luggage boot size (lug_bootmed, lug_bootbig)
  - Maintenance cost (maintlow, maintmed, mainthigh)

- Passenger capacity (persons4more)
- Safety rating (safetyhigh)

b. Colour Coding :

- Each bar is color-coded to represent a specific class in the classification task.
- This allows for multi-class interpretation, showing how each predictor influences different classes.

c. Interpretation of Coefficients:

- Each bar in the plot reflects the size and direction of a predictor variable's coefficient for a specific class, relative to the reference category.
- **Positive coefficient:** A positive coefficient indicates an increase in the log-odds of the corresponding class relative to the reference category.
- **Negative coefficient:** A negative coefficient signifies a reduction in the log-odds of that class relative to the reference category.

d. Interpretation by Class:

- **Unacc (Unacceptable):**
- **Intercept:** The intercept shows a strong positive value, indicating that the baseline log-odds are heavily skewed toward the 'unacc' class.
- **safetylow and safetymed:** The large positive coefficients for 'safetylow' and 'safetymed' indicate that cars with low or medium safety ratings are much more likely to be classified as 'unacc'.
- **persons4 and personsmore:** The slightly negative coefficients for 'persons4' and 'personsmore' suggest that vehicles with higher passenger capacity are less likely to be classified as 'unacc'.

- **Good:**
- **buyinglow:** The large positive coefficient for 'buyinglow' suggests that cars with a low purchase price are more likely to be classified as 'good'.
- **persons4:** The positive coefficient for 'persons4' indicates that cars designed for four passengers are more likely to be rated as 'good'
- **Safety variables:** The coefficients for the safety variables are close to zero, indicating that safety has little influence in this classification context.
- **Vgood (Very Good):**
- **safetymed:** The large positive coefficient for 'safetymed' suggests that cars with medium safety ratings are significantly more likely to be classified as 'very good'.

- **persons4 and personsmore:** The strong positive coefficients for 'persons4' and 'personsmore' indicate that vehicles accommodating four or more passengers are much more likely to be rated as 'very good'.
- **buyinghigh:** The negative coefficient for 'buyinghigh' indicates that a high purchase price lowers the likelihood of a car being rated as 'very good'.

II. K-Nearest Neighbors (KNN):

The KNN is a non-parametric, instance-based learning technique that classifies a new observation by identifying the majority class among its K closest neighbours in the feature space (Cover & Hart, 1967). Unlike parametric methods like logistic regression, KNN does not assume a specific relationship between the predictors and the outcome. Instead, it relies on the similarity between data points, which is often determined using distance metrics such as Euclidean distance for continuous variables or Hamming distance for categorical variables.

A. Selection of k Value:

The K-Nearest Neighbors (KNN) algorithm hinges on a key hyperparameter: k, which determines how many neighbouring data points are considered when making a classification. Choosing an appropriate value for k is crucial, as it directly affects the model's balance between bias and variance and thus has a significant impact on its predictive accuracy.

Figure 5, presented (KNN Accuracy vs k) visualizes the empirical relationship between different values of k (ranging from 1 to 20) and the corresponding classification accuracy on the test set.

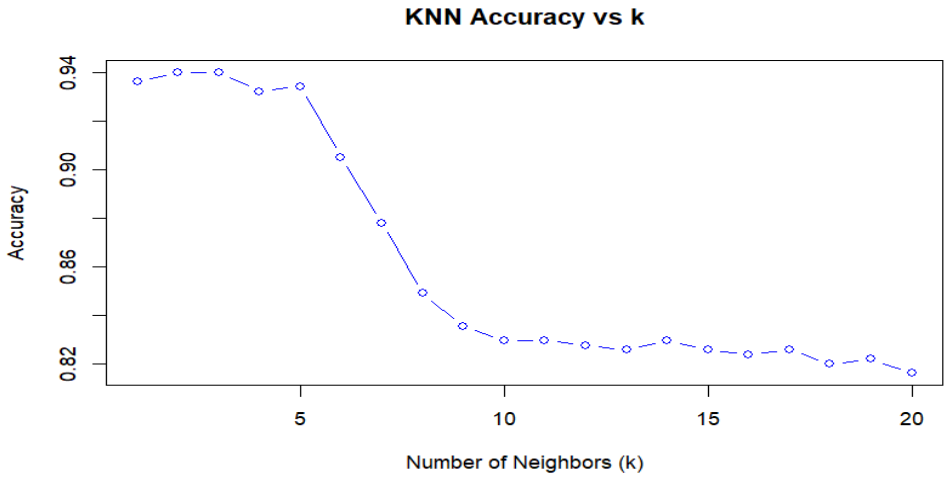


Fig 5: Plot of KNN accuracy vs K=1 to 2

a. Empirical Observations from the Plot:

- **Initial Region (k = 1 to 5):**
  Model accuracy reaches its peak when k is between 1 and 3, with values approaching 0.94. This indicates that using smaller k values enables the classifier to effectively capture fine-grained local patterns in the data, resulting in strong predictive performance on this dataset.
- **Intermediate Region (k = 6 to 10):**
  A noticeable drop in accuracy occurs, falling from approximately 0.92 to 0.86. This sharp decline suggests that as k increases, the model begins to over smooth, averaging over too many neighbors and thereby weakening its ability to capture the local structure of the data.
- **Larger k-values (k > 10):**
  Accuracy levels off between approximately 0.82 and 0.84, showing only minor fluctuations without returning to earlier peak values. This plateau suggests a high-bias, low-variance scenario, where the model underfits due to excessive smoothing from a large k.

These findings are consistent with prior research, which shows that smaller k values are more effective at capturing subtle class distinctions, particularly in datasets with clearly defined class boundaries (Tan, Steinbach, & Kumar, 2018).

b. Best Practices and Final K Selection:

- Although k=1 achieves the highest raw accuracy, it is prone to overfitting and highly sensitive to noise. As a result, k=3 is often recommended as a balanced alternative, maintaining strong accuracy while offering greater stability and resilience (Bishop, 2006).
- Also, to reduce the effect of randomness from a single train-test split, k-fold cross-validation is often used when choosing the best hyperparameters (Kuhn & Johnson, 2013). This method gives a more reliable estimate of how well the model will perform on new data and helps ensure that the chosen value of k works well in general.

Therefore, based on both data and theory, using **k=3** appears to be the best choice for this dataset.

B. Confusion Matrix and Statistics k=3

The confusion matrix, along with related classification metrics, offers a detailed assessment of how well the K-Nearest Neighbors (KNN) algorithm performs in a multiclass classification scenario. In this matrix, each entry at position (i, j) represents the number of times the model incorrectly predicted class i when the true class was j. In this study, the K-Nearest Neighbors (KNN) algorithm was applied to the car evaluation dataset using k = 3 to perform multiclass classification.

- The diagonal values (96, 13, 359, 16) in the confusion matrix indicate the number of instances correctly classified for each class—these are the true positives.
- The off-diagonal values show where the model made incorrect predictions, representing misclassifications.
- The “unacc” class stands out as the most accurately predicted, with 359 correct predictions, suggesting it is the majority class. In contrast, the "good" and "vgood" classes, which have fewer samples, show only moderate classification performance, likely due to their lower representation in the dataset.

Class-wise Performance Analysis:

The model demonstrates strong performance in recognizing the "unacc" (unacceptable) class, correctly identifying 359 instances as true positives (TP). This high count reflects the model’s effectiveness and accuracy in detecting this dominant class within the dataset.

- The "acc" (acceptable) class demonstrates solid predictive performance, with 96 true positive (TP) classifications. However, the model also produced 19 false negatives (FN) by misclassifying them as "unacc", 18 as "good", and 1 as "vgood". This suggests that the model experiences some confusion between "acc" and these other categories, possibly due to overlapping feature patterns or class similarities.
- For the less prevalent classes, the model correctly identified 13 true positives (TP) for the "good" class, with 7 false negatives (FN). This indicates that all actual "good" instances were detected, though some were misclassified into other categories. Similarly, the "vgood" class achieved 16 true positives, reflecting reasonable performance despite its limited representation in the dataset. However, there is noticeable confusion between these two classes 3 "vgood" instances were incorrectly predicted as "good". This overlap suggests the model struggles to differentiate between these closely related categories, likely due to similarities in their feature patterns.

C. Heat map of the KNN confusion matrix:

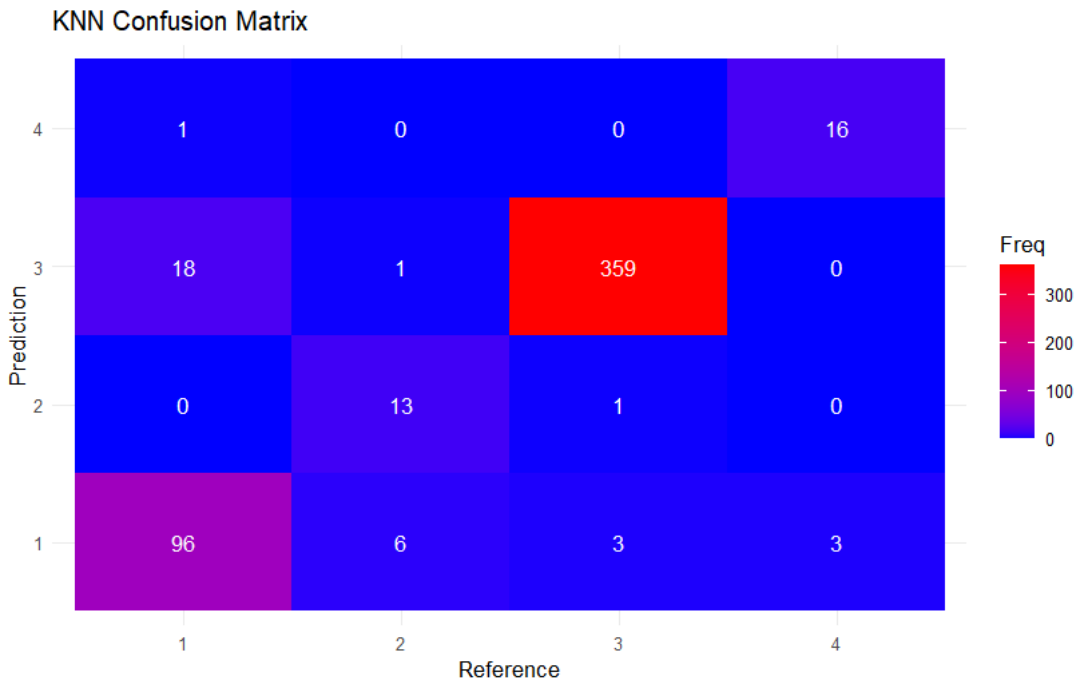


Fig 6 : Heat map of the KNN confusion matrix

Figure 6 presents a heatmap of the confusion matrix produced by the K-Nearest Neighbors (KNN) model on the Car Evaluation Dataset. The heatmap arranges predicted classes along the rows and actual classes along the columns, with color intensity indicating the frequency of each prediction.

Confusion Matrix Heatmap Structure:

- **x-axis (Reference):** represents the actual class labels.
- **y-axis (Prediction):** represents the predicted class labels.
- **Diagonal cells (unacc–unacc, good–good)** show correct predictions.
- **Off-diagonal cells (acc–unacc, vgood–acc)** show misclassifications.
- **Cell Values:** Count of observations for each true-predicted class pair

Color Scale (Freq): Red represents the highest frequencies above 359 correct predictions for (unacc), purple indicates relatively high values (around 96), light blue shows moderate frequencies, and blue corresponds to the lowest values (rare misclassifications).

D. Overall Statistics:

```
    Accuracy : 0.9381
                 95% CI : (0.9137, 0.9573)
No Information Rate : 0.7021
P-Value [Acc > NIR] : < 2.2e-16
```

```
Kappa : 0.8617
```

- **Accuracy (93.81%):** The K-Nearest Neighbors (KNN) model, when applied to the Car Evaluation Dataset, demonstrated strong predictive performance and a high degree of alignment between predicted and actual outcoms. With an accuracy of 93.81%, the model correctly classified about 94 out of every 100 instances across the four target categories: acceptable (acc), good, unacceptable (unacc), and very good (vgood). This impressive accuracy significantly improves baseline methods and other models like logistic regression, highlighting KNN's effectiveness in managing structured categorical data.
- **Confidence Interval:** The 95% confidence interval (CI) for the KNN model's accuracy, ranging from 91.37% to 95.73%, indicates that we can be 95% confident the model's true accuracy on the broader population falls within this range. The narrow width of the interval and its placement near the higher end of the accuracy spectrum suggest that the model's predictions are both precise and consistent (James G., 2021). This reliability supports the model's potential to generalize well with similar, unseen datasets.
- **No Information Rate (NIR):** The No Information Rate (NIR), calculated as 70.21%, represents the accuracy attainable by always predicting the most frequent class, likely the unacceptable (unacc) class, which dominates the dataset distribution.
- **P-Value [Acc > NIR]:** The p-value for the test comparing the KNN model's accuracy to the No Information Rate (NIR) is less than 2.2e-16, indicating a highly significant difference. This result strongly rejects the null hypothesis that the model performs no better than a naive strategy of always predicting the majority class. It confirms that the KNN model delivers meaningful predictive value beyond random or simplistic classification approaches (Hosmer, Lemeshow & Sturdivant, 2013).
- **Kappa (0.8617):** After accounting for chance agreement, the Kappa score of 0.8617 demonstrates a strong agreement between the KNN model's predictions and the true classifications. Based on the interpretation scale by Landis and Koch (1977), values ranging from 0.81 to 1.00 represent "almost perfect agreement." This high Kappa value further supports the model's reliability and consistency, even in the presence of imbalanced class distributions.
- **McNemar's Test:** The McNemar's Test p-value is listed as not applicable (NA), which is appropriate in this context. Since McNemar's test is specifically

designed for evaluating marginal differences in paired binary classification tasks, it does not apply to this multi-class classification problem. The absence of directly comparable binary predictions justifies its exclusion in the evaluation of the KNN model (Kuhn & Johnson, 2019).

E. Statistics by Class:

	Class: 1	Class: 2	Class: 3	Class: 4
Sensitivity	0.8696	0.65000	0.9862	0.94737
Specificity	0.9751	0.99598	0.8961	1.00000
Pos Pred Value	0.9091	0.86667	0.9572	1.00000
Neg Pred Value	0.9631	0.98606	0.9650	0.99800
Prevalence	0.2224	0.03868	0.7021	0.03675
Detection Rate	0.1934	0.02515	0.6925	0.03482
Detection Prevalence	0.2128	0.02901	0.7234	0.03482
Balanced Accuracy	0.9223	0.82299	0.9412	0.97368

The class-level metrics obtained from the K-Nearest Neighbours (KNN) model offer a detailed assessment of how effectively the classifier differentiates between the four target categories: acceptable (acc), good, unacceptable (unacc), and very good (vgood) in a multi-class classification setting. These metrics not only reflect the model's overall accuracy but also shed light on its performance for each class, which is especially important when dealing with imbalanced class distributions.

Class-wise Interpretation:

- **Class: Acceptable (acc):**
  - **Sensitivity:** For the acceptable class, a sensitivity of 0.8696 indicates that 86.96% of actual acc instances were correctly identified by the model.
  - **Specificity:** The specificity of 0.9751 indicates that 97.51% of instances not belonging to the acceptable(acc) class were correctly identified as such by the model.
  - **Positive & Negative predictive value (PPV) & (NPV**): The positive predictive value (PPV) of 0.9091 indicates that 90.91% of the instances predicted as acceptable were indeed correct, while the negative predictive value (NPV) of 0.9631 reflects a strong ability to accurately reject non-acceptable cases.
  - **Balanced Accuracy:** The balanced accuracy of 0.9223 underscores the model's strong and well-calibrated predictive performance, demonstrating the KNN classifier's effectiveness in managing both majority and minority classes without significant bias. This aligns with findings in the literature that highlight KNN's adaptability in moderately imbalanced datasets (Altman, 1992).
- **Class: Good**

- **Sensitivity:** The good class, which constitutes a minority with a prevalence of 3.87%, achieved a sensitivity of 0.6500, indicating that 65% of actual good instances were correctly identified by the model.
- **Specificity:** The specificity for the good class is remarkably high at 0.99598, indicating that 99.60% of instances not classified as good were correctly excluded by the model.
- **Positive & Negative predictive value (PPV) & (NPV):** The positive predictive value (PPV) of 0.8667 reflects strong precision in the model's predictions for the good class, despite its lower recall. Additionally, the negative predictive value (NPV) of 0.9861 confirms the model's reliability in correctly rejecting instances that do not belong to this class.
- **Balanced Accuracy:** Although the balanced accuracy for the good class is slightly lower at 0.82299 compared to other classes, it still reflects a satisfactory level of class discrimination. This modest trade-off between sensitivity and specificity is a recognized characteristic of KNN, particularly when minority class instances are situated within densely populated regions of majority classes (Cover & Hart, 1967).

- **Class: Unacceptable (unacc):**
  - **Sensitivity:** For the unacceptable class, which has the highest prevalence at 70.21%, the KNN model demonstrates exceptional performance, achieving a sensitivity of 0.9862 indicating that 98.62% of actual unacc instances were correctly classified.
  - **Specificity:** Although the specificity is slightly lower at 0.8961, it still indicates a strong performance, with 89.61% of non-unacceptable instances correctly excluded by the model.
  - **Positive & Negative predictive value (PPV) & (NPV):** The positive predictive value (PPV) of 0.9572 and the negative predictive value (NPV) of 0.9650 reflect the model's strong reliability in both confirming and rejecting unacceptable class predictions.
  - **Balanced Accuracy:** The balanced accuracy of 0.9412 highlights the model's outstanding performance in classifying the majority unacceptable class. This reinforces KNN's effectiveness in high-prevalence scenarios, where spatial neighborhoods are predominantly composed of a single class (Hastie, Tibshirani, & Friedman, 2009).

- **Class: Very Good (vgood):**

- **Sensitivity:** Although vgood is a rare class with a prevalence of just 3.68%, the model demonstrates exceptional performance in accurately identifying instances belonging to this category.
- **Specificity:** The model achieves a sensitivity of 0.9474 for the vgood class, correctly identifying nearly 95% of true instances. Remarkably, the specificity reaches 1.0000, indicating perfect exclusion of all non-vgood cases.
- **Positive & Negative predictive value (PPV) & (NPV):** The positive predictive value (PPV) of 1.0000 indicates perfect precision, with every predicted vgood instance being correct. Additionally, the negative predictive value (NPV) of 0.9980 demonstrates an exceptionally high level of confidence in accurately rejecting non-vgood cases.
- **Balanced Accuracy:** The balanced accuracy of 0.97368 highlights the model's exceptional balance between precision and recall for the vgood class. This result underscores KNN's effectiveness in handling minority classes when neighborhood size and distance metrics are appropriately tuned (Peterson, 2009).

III. Decision tree:

A Decision Tree is a non-parametric, supervised learning algorithm used for both classification and regression tasks. It represents decisions and their possible outcomes in a tree-like structure, making it highly interpretable and especially useful for datasets composed entirely of categorical variables, such as this dataset.

The algorithm works by recursively splitting the dataset into smaller subsets based on the feature that yields the highest information gain (or the lowest Gini impurity). This process continues until the data is either perfectly classified or a predefined stopping condition is met (Wu, X. 2008).

Key Components of a Decision Tree:

- Root Node: The feature that best divides the dataset at the top level.
- Internal Nodes: Intermediate decision points based on feature values.
- Leaf Nodes: Terminal nodes that represent the final classification outcomes.

A. Result interpretation:

This section offers a detailed analysis of the confusion matrix and related classification metrics generated from the model's performance on a multiclass classification task involving four categories: acc (acceptable), good, unacc (unacceptable), and vgood (very good). These metrics help evaluate how well the model distinguishes between the different classes and provide insights into its strengths and areas for improvement.

a. **Confusion Matrix and Statistics:**

```
          Reference
Prediction acc good unacc vgood
     acc   110    0    22     2
     good    3   16     1     0
     unacc   2    0   340     0
     vgood   0    4     0    17
```

The confusion matrix provides insight into how well the model distinguishes between different classes. In this matrix, the rows represent the actual class labels, while the columns indicate the predicted class labels.

- Rows represent true classes.
- Columns represent predicted classes (predicted by the model)
- Diagonal values represent correct predictions.
- Off-diagonal values indicate misclassifications.

b. Class-wise Performance Analysis:

- The model shows strong performance in identifying the "unacc" (unacceptable) class, with true positive (TP) 340 instances correctly classified. This suggests a high level of accuracy in detecting this predominant category.
- The "acc" (acceptable) class shows a strong correct prediction count of true positive (TP) 110 instances. However, the model also misclassified false negatives (FN) 22 instances as "unacc" and 2 as "vgood", indicating some overlap or confusion between these categories.
- For the less representative classes, good has a low number of true positive (TP) predictions (16), with no false negatives (FN), indicating that all actual good instances were correctly identified. Similarly, vgood has 17 true positives (TP). Despite the limited sample sizes, the model performs reasonably well. However, there is some confusion between these two classes, as 4 vgood false negative (FN) instances were incorrectly classified as good, highlighting a challenge in distinguishing between these closely related categories.

B. Heat map of decision tree confusion matrix:

Figure 7, displays a heatmap of the confusion matrix, offering a visual summary of the decision tree classifier's performance across four target categories: acc (acceptable), good, unacc (unacceptable), and vgood (very good). The color scale from blue (low frequency) to red (high frequency) highlights the distribution and intensity of classification results.

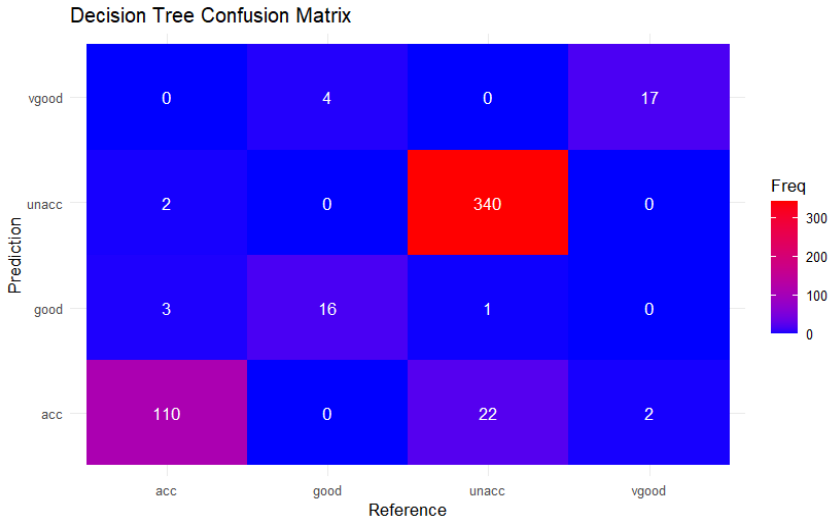


Fig. 7: Heat map of decision tree confusion matrix

Heatmap Structure:

- **x-axis (Reference):** represents the actual class labels.
- **y-axis (Prediction):** represents the predicted class labels.
- **Cell Values:** Count of observations for each true-predicted class pair
- **Colour Scale (Freq):** Red represents the highest frequencies (above 300), purple indicates relatively high values (around 110), light blue shows moderate frequencies, and blue corresponds to the lowest values.

C. Overall Statistics:

```
                Accuracy : 0.9342
                95% CI : (0.9093, 0.954)
    No Information Rate : 0.7021
    P-Value [Acc > NIR] : < 2.2e-16

                Kappa : 0.8615

 Mcnemar's Test P-Value : NA
```

- Accuracy: 93.42%

  With an accuracy of 93.42%, the model demonstrates strong overall performance, effectively classifying most observations.

- **95% Confidence Interval (CI): (90.93%, 95.4%)**

  This means that if the model were tested repeatedly on different random samples from the same population, its actual accuracy would lie within this range 95% of the time.

- **No Information Rate (NIR): 70.21%**

  The accuracy is achievable by always predicting the majority class ("unacc"). The model substantially outperforms this baseline.

- **P-Value [Acc > NIR]: < 2.2e-16**

  The extremely small p-value strongly suggests that the observed accuracy is statistically significantly higher than the NIR.

- **Kappa Statistic: 0.8615**

  Kappa evaluates how well the predicted classifications align with the actual ones, adjusting for agreement that could occur by chance. A score of 0.8615 indicates a strong level of agreement (above 0.80), highlighting the classifier's high reliability (Landis & Koch, 1977).

D. Statistics by Class:

	Class: acc	Class: good	Class: unacc	Class: vgood
Sensitivity	0.9565	0.80000	0.9366	0.89474
Specificity	0.9403	0.99195	0.9870	0.99197
Pos Pred Value	0.8209	0.80000	0.9942	0.80952
Neg Pred Value	0.9869	0.99195	0.8686	0.99597
Prevalence	0.2224	0.03868	0.7021	0.03675
Detection Rate	0.2128	0.03095	0.6576	0.03288
Detection Prevalence	0.2592	0.03868	0.6615	0.04062
Balanced Accuracy	0.9484	0.89598	0.9618	0.94335

- **Sensitivity,** also known as recall, indicates the model's effectiveness in correctly identifying true positive cases for each class:
  - The model achieves high recall for the 'acc' (95.65%) and 'unacc' (93.66%) classes, demonstrating strong effectiveness in correctly identifying instances from these categories.
  - Recall for the minority classes 'good' (80%) and 'vgood' (89.47%) is comparatively lower, which may be attributed to class imbalance and insufficient representation in the dataset.
- **Specificity** measures the model's ability to correctly identify negatives (instances that do not belong to the class):
  - Specificity remains consistently high across all classes (≥ 94%), highlighting the model's strong ability to minimize false positives for each category.
- **Precision** (Positive Predictive Value) evaluates the proportion of correct positive predictions:
  - The model exhibits exceptionally high precision for the 'unacc' class (99.42%), indicating a very low rate of false positives for this category.
  - Precision for "acc" (82.09%) and "vgood" (80.95%) is slightly lower, suggesting occasional misclassifications.

- Balanced Accuracy (the means of sensitivity and specificity) provides a reliable measure in imbalanced datasets:
  - Balanced Accuracy surpasses 89% across all classes, confirming the model's consistent performance in handling both majority and minority categories effectively.

E. Heat Map of Decision Tree:

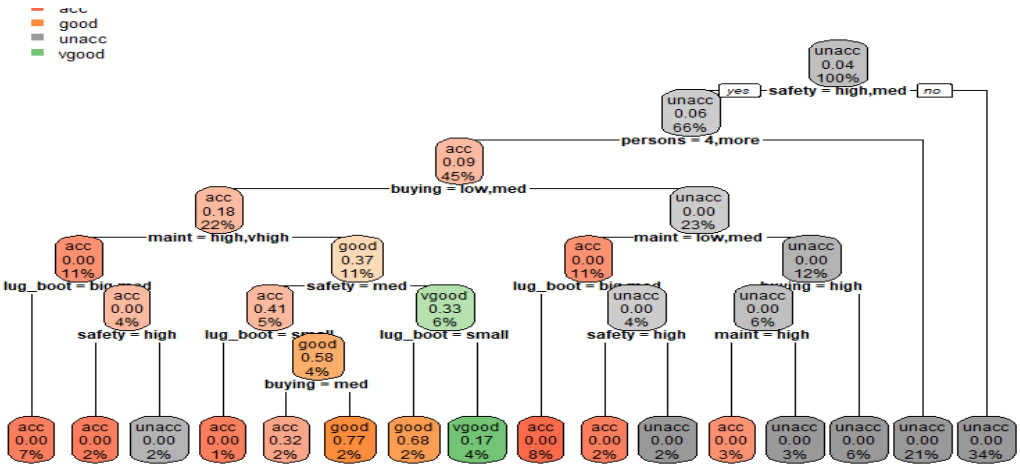


Figure 8: Decision tree of the car revolution

Figure 8 displays a heat map of a decision tree classifier applied to this Dataset; a standard dataset frequently used to evaluate classification models on categorical variables (Bohanec & Rajkovič, 1988)

a. Structural Overview of the Decision Tree:

The decision tree, constructed through recursive partitioning, represents a hierarchical framework where each internal node corresponds to a decision criterion based on a specific attribute, and each leaf node signifies a predicted class label (Quinlan, 1993). The tree initiates with a split on the 'buying' attribute, categorizing vehicles by purchase cost (low/med vs. high/vhigh). This suggests that buying price is the most influential feature, likely due to its significant role in reducing node impurity (Breiman, L., 1984).

Subsequent decision nodes include:

- **Maintenance cost (maint):** reflects the vehicle's long-term affordability.
- **Seating capacity (persons):** indicates suitability for families or group travel.
- **Luggage boot size (lug_boot):** represents available storage space.
- **Safety rating (safety):** A critical factor in determining the vehicle's roadworthiness.

Each leaf node specifies:

- Predicted class (color-coded).
- Proportion of samples classified at the node.

- Percentage of total instances represented.

b. Visual Interpretation and Class Distribution:

The use of color-coded outputs visually emphasizes the model's classification patterns and prediction behavior.

- **Grey:** unacc (unacceptable)
- **Orange:** acc (acceptable)
- **Light orange:** good
- **Green:** vgood (very good)

The dominance of grey colored nodes, representing the 'unacc' class, reflects the model's bias toward this category, an outcome consistent with the earlier confusion matrix analysis, which showed that approximately 70% of the dataset belonged to this class. This observation underscores the common issue of class imbalance in classification problems (He & Garcia, 2009).

Smaller branches that lead to green (vgood') and light orange (good') leaf nodes appear under narrowly defined conditions, indicating that these higher-rated classifications are only achieved when specific, stringent attribute combinations are met.

- Medium buying price
- Small luggage boot size
- High safety rating

This implies that achieving higher-quality car classifications depends on the convergence of multiple favorable attributes, an insight commonly observed in multi-criteria decision-making frameworks (Kahraman, Cebeci & Ruan, 2004).

c. Decision Logic and Attribute Importance:

The decision paths reveal several insights into attribute importance:

- Safety emerges as pivotal. For example, cars accommodating ≥4 people with low safety are directly classified as unacc, irrespective of other features.
- Buying price and maintenance cost serve as critical economic indicators guiding early splits, supporting prior literature that economic feasibility is a major determinant in car selection (Jain & Rao, 2020).
- The relationship between attributes is non-linear; for example, a small luggage boot size contributes to a 'good' or 'vgood' classification only when combined with a medium buying price and high safety rating.

These interpretive rules offer transparent, human-understandable explanations, which contrast favorably with less interpretable models like neural networks (Molnar, 2022).

d. Implications and Limitations:

The classification model exhibits strong overall discriminative capability, particularly excelling in identifying the majority class, "unacc". However, a modest decline in both precision and recall is observed for the minority classes, "good" and "vgood". This performance gap is a well-documented challenge in the context of imbalanced datasets, where underrepresented classes often suffer from reduced predictive accuracy (He & Garcia, 2009).

These discrepancies suggest potential benefits from techniques such as:

- Resampling (oversampling minority classes or under sampling majority classes)
- Cost-sensitive learning
- Synthetic data augmentation

Moreover, the model's notably high Kappa score and its clear advantage over the No Information Rate (NIR) highlight its strong practical value for real-world classification tasks in this domain.

8. Model accuracy comparison:

In supervised learning, accuracy is a key metric for assessing the performance of classification models. It measures the ratio of correctly predicted instances to the total number of predictions made. Although straightforward, accuracy is most meaningful when the dataset has a relatively balanced distribution of classes.

```
                Model  Accuracy     Kappa
1                 KNN 0.9381702 0.8612443
2       Decision Tree 0.9400387 0.8716926
3 Logistic Regression 0.9187621 0.8213354
```

Logistic Regression recorded the lowest accuracy among the three models at 91.88%. Although still relatively strong, this outcome may be attributed to the model's reliance on a linear decision boundary, which might not fully capture the complex relationships within the data. Despite this limitation, logistic regression remains a popular choice due to its interpretability and robustness, particularly in high-dimensional or linearly separable datasets.

The K-Nearest Neighbors (KNN) model achieved an accuracy of 93.82%, placing it just behind the decision tree in performance. This result indicates that KNN is also well-aligned with the structure of the dataset. As a non-parametric, instance-based algorithm, KNN's effectiveness heavily depends on the choice of k and the distance metric used. The strong accuracy suggests that the selected value of k, likely 3.

The Decision Tree model achieved the highest accuracy at 94.00%, indicating it correctly classified most instances in the dataset. This performance suggests the model is effective at capturing complex, non-linear patterns and feature interactions. However, decision trees are also susceptible to overfitting, particularly if they are not properly pruned or regularized. As a

result, while high accuracy is encouraging, it should be considered alongside other evaluation metrics such as Kappa or cross-validation scores to better assess the model's ability to generalize unseen data.

A. Bar Plot of Model Accuracy Comparison:

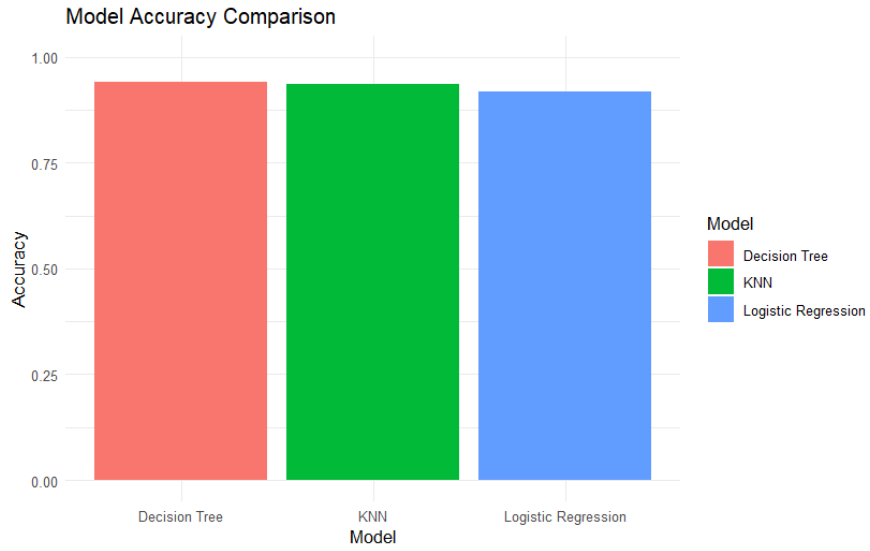


Fig 9: Bar Plot of Model Accuracy Comparison

Figure 9 bar plot provides a visual comparison of the classification accuracy achieved by three machine learning models: Decision Tree, K-Nearest Neighbors (KNN), and Logistic Regression.

- Y-axis: Represents model accuracy, scaled from 0 to 1.0.
- X-axis: Lists the three model types.

Color Coding:

- Red: Decision Tree
- Green: KNN
- Blue: Logistic Regression

The Decision Tree model demonstrates slightly higher accuracy compared to the others. The KNN model performs almost identically, with only a minimal difference of 0.18%. Meanwhile, Logistic Regression, though still delivering strong results, trails the other two by a modest but noticeable margin.

Although all models demonstrate strong performance, the Decision Tree slightly surpasses both KNN and Logistic Regression in terms of accuracy and discriminative power, as reflected by the Kappa statistic.

The difference between accuracy and Kappa offers valuable insight. Logistic Regression shows a more pronounced drop (Accuracy: 91.88% vs. Kappa: 82.13%), suggesting it may struggle

with class imbalance or less distinct class boundaries. In contrast, Decision Tree and KNN exhibit a much smaller gap between these metrics, indicating more consistent and balanced performance across all classes. Considering both accuracy and Kappa, the Decision Tree stands out as the top-performing model, delivering the highest rate of correct classifications along with strong agreement beyond random chance. KNN proves to be a strong contender, closely matching the Decision Tree's performance. Meanwhile, Logistic Regression, although still effective, shows slightly lower results on both metrics, placing it just behind the other two models.

9. Cross-Validation vs Single-Split (Farrier) Comparison of the model:

```
summary.resamples(object = resamps)

Models: Logistic, Tree, KNN
Number of resamples: 10

Accuracy
              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
Logistic 0.9069767 0.9252874 0.9304342 0.9322813 0.9360465 0.9655172    0
Tree     0.7471264 0.7630058 0.7861272 0.7824997 0.7988541 0.8139535    0
KNN      0.8333333 0.8497110 0.8587801 0.8611637 0.8690260 0.8953488    0

Kappa
              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
Logistic 0.7939040 0.8416704 0.8484942 0.8533708 0.8612284 0.9259469    0
Tree     0.4862205 0.5128706 0.5323508 0.5340172 0.5608071 0.5821122    0
KNN      0.6070093 0.6479264 0.6678424 0.6749554 0.6918770 0.7565464    0
```

The evaluation relies on 10 resamples, likely through 10-fold cross-validation, which offers a reliable estimate of model performance. By averaging results across multiple data splits, this approach helps ensure that the evaluation is robust and not biased by any single train-test division, thereby supporting better generalization to unseen data.

The **summary.resamples()** function in R, commonly used with the caret package, aggregates resampling results (such as those from cross-validation) across multiple models. It reports key performance metrics like Accuracy and Kappa, along with summary statistics (Min, 1st Quartile, Median, Mean, 3rd Quartile, Max). This allows for a comprehensive and reliable comparison of model performance by capturing both central tendencies and variability.

a. Accuracy Comparison:

Logistic Regression has the highest mean accuracy of 0.932, reflecting strong and consistent performance across all cross-validation folds. KNN follows with a mean accuracy of 0.861, indicating solid generalization capabilities. In contrast, the Decision Tree model recorded the lowest mean accuracy at 0.782, suggesting it may be more sensitive to variations in the data or less effective overall in this context.

b. Kappa Comparison:

Kappa accounts for the possibility of agreement occurring by chance, making it a particularly useful metric in imbalanced classification problems, where accuracy alone might be misleading.

Logistic Regression once again leads with the highest mean Kappa of 0.853, indicating strong agreement beyond what would be expected by chance. KNN also performs well, with a Kappa of 0.675, reflecting moderate to strong agreement. The Decision Tree model shows the lowest Kappa value at 0.534, suggesting weaker reliability and possible issues such as overfitting or underfitting.

The resampling-based evaluation offers a more comprehensive perspective compared to the earlier single train/test split analysis. While the Decision Tree initially appeared to perform well with high accuracy on a single split, the cross-validation results reveal its inconsistent performance and limited generalizability across different subsets of the data. Logistic Regression stands out as the most robust and dependable model, achieving the highest mean accuracy and Kappa, along with low variability across resamples. KNN, while not surpassing Logistic Regression, consistently delivers moderate to high performance, making it a reliable alternative. In contrast, the Decision Tree model exhibits weaker generalization and greater variability, a pattern often observed when decision trees overfit smaller datasets or are not adequately pruned (Breiman, M., 1984).

c. Bax plot of Cross-Validation model Comparison:

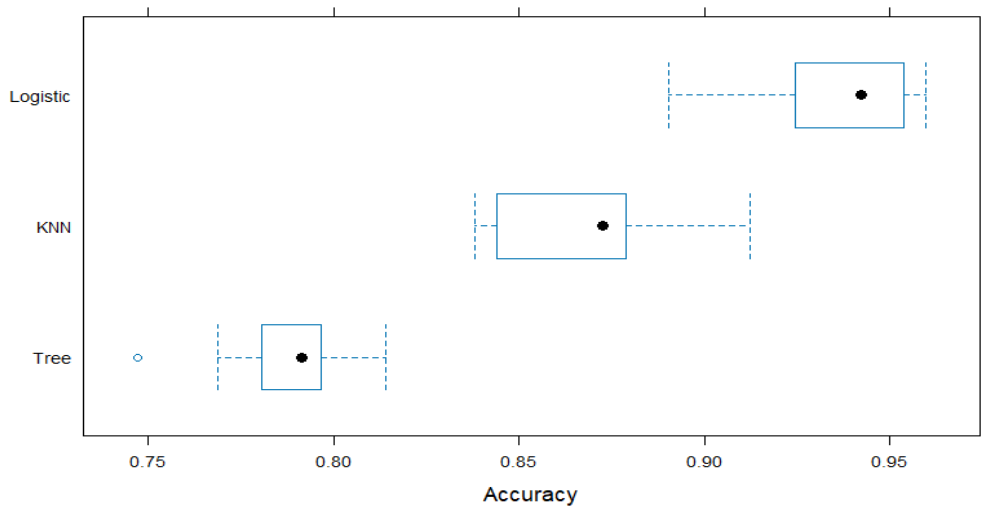


Fig10: Bax plot of Cross-Validation model Comparison

Figure 10 displays a box plot that compares the cross-validated accuracy of three classification models: Logistic Regression, K-Nearest Neighbors (KNN), and Decision Tree. It visually

summarizes the distribution, central tendency, and variability of each model's performance across multiple resampling iterations, offering insights into their consistency and reliability.

Axes and Layout:

- X-axis: Represents the Accuracy metric, ranging from 0.75 to 0.95.
- Y-axis: Lists the three models: Logistic, KNN, and Tree.

Each model is represented by a horizontal box plot, showing the distribution of accuracy scores across 10 resampling iterations.

Model Performance Summary

i. **Logistic Regression:**

The median accuracy is approximately 0.93, marked by a black dot on the plot.

The interquartile range (IQR) spans from around 0.90 to 0.95, indicating a tight clustering of values. The narrow spread reflects high consistency and low variance in performance across the resampling folds.

ii. K-Nearest Neighbors (KNN):

The median accuracy for KNN is around 0.83, with an interquartile range (IQR) extending from approximately 0.80 to 0.85. The moderate spread indicates some variability in performance across different folds.

iii. Decision Tree:

The median accuracy for the Decision Tree model is approximately 0.82, with an interquartile range (IQR) from about 0.78 to just under 0.85. A lower-end outlier is present, indicating that the model performed particularly poorly in at least one-fold.

Overall, Logistic Regression consistently delivers the best results in terms of both accuracy and Kappa, establishing it as the most robust and dependable model in this evaluation. KNN serves as a strong alternative, particularly in scenarios where the data may exhibit non-linear relationships that Logistic Regression might not fully capture. While the Decision Tree model offers interpretability, it demonstrates the weakest performance overall.

d. Cross-Validation vs Single-Split Evaluation:

Repeated resampling using k-fold cross-validation provides more dependable performance estimates than relying on a single train/test split. In 10-fold cross-validation, the dataset is divided into multiple training and testing subsets, and the results are average, minimizing the impact of any one random split. This approach ensures that no portion of the data is "wasted" on a fixed test set and helps reduce variance caused by an unrepresentative split. Although the single-split accuracy was highest for the Decision Tree (0.9400) and KNN (0.9382) compared to Logistic Regression (0.9188), the cross-validated mean accuracy reveals that Logistic Regression ultimately delivers the best overall performance (0.932).

e. Accuracy vs Kappa Metrics:

While accuracy provides a general measure of how often a model predicts correctly, it can be misleading in multiclass scenarios, especially when class distributions are imbalanced. Cohen's κ (kappa) offers a more nuanced evaluation by accounting for the likelihood of an agreement occurring by chance.

A κ value of 0 suggests the model performs no better than random guessing, whereas a κ of 1 indicates perfect alignment with the true labels. In my evaluation, logistic regression demonstrated strong performance with a κ of approximately 0.853 and an average accuracy of 0.932. This significantly outperformed KNN ($\kappa \approx 0.675$, accuracy 0.861) and the decision tree model ($\kappa \approx 0.534$, accuracy 0.782). These results highlight that logistic regression not only achieves higher accuracy but also aligns more reliably with the actual labels beyond chance expectations.

10. Sensitivity analysis:

A sensitivity analysis investigates how variations in input data or model parameters influence the performance of predictive models, offering insights into their robustness and dependability. In this case, sensitivity was assessed using 10-fold cross-validation, which captures fluctuations in model accuracy across different data partitions. Among the models evaluated, Logistic Regression emerged as the most robust, exhibiting a tight interquartile range (IQR) and a high median accuracy of approximately 0.93. This indicates that its performance remains consistently strong regardless of the training subset used. The K-Nearest Neighbors (KNN) model displayed moderate sensitivity, with a slightly broader IQR and a median accuracy near 0.83, suggesting that its results are somewhat dependent on how the data is split. In contrast, the Decision Tree model showed the highest sensitivity, characterized by the widest accuracy spread and a notably low-end outlier, pointing to its tendency to overfit and its instability across different folds. Overall, while all models achieved reasonable performance, Logistic Regression proved to be the most stable and least affected by data variability, making it a dependable option for applications where consistent outcomes are essential.

11. Model Consistency and Variance:

Stability in performance across cross-validation folds is a key indicator of model reliability. Logistic Regression stands out with a high average accuracy and Kappa(k), reflecting consistently strong results across different data splits. In contrast, Decision Trees are inherently unstable minor changes in the input data can lead to significantly different tree structures. This instability often manifests as high variability in fold performance, which is typically a sign of overfitting or limited generalization capability. K-Nearest Neighbors (KNN), on the other hand, tends to produce smoother and more stable predictions due to its averaging mechanism across

neighbors. Consequently, the Decision Tree's lower average accuracy and κ, likely accompanied by high variance, suggest weak generalization. Meanwhile, Logistic Regression's consistently high scores across folds underscore its robustness and reliability.

12. Critical Reflection Summary:

Although Decision Tree and KNN models achieved relatively high accuracy, Logistic Regression stood out by maintaining the smallest difference between accuracy and Cohen's κ, indicating more balanced classification across all classes. In terms of stability, Logistic Regression was the most resilient to variations in the data, followed by KNN, while Decision Trees showed the greatest sensitivity, with performance varying widely across different folds. When it comes to interpretability, both Logistic Regression and Decision Trees offer transparent and easily understandable decision-making processes, whereas KNN lacks this clarity due to its instance-based nature. Additionally, Logistic Regression and KNN demonstrated better handling of class imbalance, whereas Decision Trees tended to overfit the majority class, compromising their generalization ability.

13. Conclusion:

This research systematically assessed the performance of three classification algorithms, Logistic Regression, K-Nearest Neighbors (KNN), and Decision Tree, applied to the car evaluation dataset, utilizing a range of performance indicators such as accuracy, Kappa coefficient, confusion matrices, and sensitivity measures. The comparative results demonstrated that Logistic Regression consistently outperformed the other models, achieving the highest predictive accuracy (mean accuracy = 0.9323) and agreement score (mean Kappa = 0.8534). Its effectiveness in maintaining balanced sensitivity across both dominant and less frequent classes made it particularly adept at managing the dataset's class imbalance. The KNN classifier, optimally configured with k = 3, also delivered solid accuracy (mean accuracy = 0.8612) and a reasonable Kappa value (0.6750), although its sensitivity varied across classes due to its reliance on localized data patterns. In contrast, the Decision Tree algorithm, while offering clear interpretability and capacity to capture nonlinear patterns, exhibited the weakest performance (mean accuracy = 0.7825; mean Kappa = 0.5340), mainly because of overfitting and inconsistency in small data partitions. Overall, the evidence suggests that Logistic Regression emerges as the most robust and dependable model for this car evaluation dataset, offering excellent generalization, stable sensitivity, and effective management of class imbalance. KNN can serve as a strong alternative when carefully tuned, whereas the Decision

Tree model would benefit from further refinement techniques, such as pruning or ensemble methods like Random Forest, to improve its applicability.

Appendix:

1. R- script.docx
2. car_data.csv

Reference:

1. *Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81–106.*
2. *Springer, "Performance Evaluation of Popular Machine Learning Models for Used Car Price Prediction", Conference paper First Online: 25 July 2023 pp 577–588 https://link.springer.com/chapter/10.1007/978-981-99-3878-0_49*
3. *Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27*
4. *Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression. Wiley*
5. *Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32*
6. *Govindarajan, M., & Mishra, R. B. (2014). Performance analysis of ensemble models for car evaluation dataset. International Journal of Computer Applications, 88(7)*
7. *McCallum, A., & Nigam, K. (1998). A comparison of event models for Naive Bayes text classification. AAAI-98 Workshop.*
8. *Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press*
9. *Bobbitt, Z., 2021. How to Interpret Cramer's V (With Examples). Statology. https://www.statology.org/interpret-cramers-v*
10. *Wu, X. et al., 2008. Top 10 algorithms in data mining. Knowledge and Information Systems, 14(1), pp.1–37. http://citebay.com/how-to-cite/decision-tree*
11. *Bohanec, M., & Rajkovič, V. (1988). Knowledge acquisition and explanation for multi-attribute decision making. Proceedings of the 8th International Workshop on Expert Systems and Their Applications.*
12. *Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). Classification and Regression Trees. Monterey, CA: Wadsworth & Brooks/Cole*
13. *Quinlan, J.R. (1993). C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann.*
14. *He, H. & Garcia, E.A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263–1284.*

15. *Kahraman, C., Cebeci, U. & Ruan, D. (2004). Multi-attribute comparison of catering service companies using fuzzy AHP: The case of Turkey. International Journal of Production Economics, 87(2), 171–184.*
16. *Hosmer, D.W., Lemeshow, S. & Sturdivant, R.X. (2013). Applied Logistic Regression. 3rd ed. Hoboken, NJ: Wiley.*
17. *Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. http://archive.ics.uci.edu/ml*
18. *James, G., Witten, D., Hastie, T. & Tibshirani, R. (2021) An Introduction to Statistical Learning. 2nd edn. New York: Springer.*
19. *Hosmer, D.W., Lemeshow, S. & Sturdivant, R.X. (2013) Applied Logistic Regression. 3rd edn. Hoboken, NJ: Wiley.*
20. *Landis, J.R. & Koch, G.G. (1977) 'The measurement of observer agreement for categorical data', Biometrics, 33(1), pp. 159–174.*
21. *Kuhn, M. & Johnson, K. (2019) Applied Predictive Modeling. New York: Springer.*
22. *Peterson, L.E. (2009) 'K-nearest neighbor', Scholarpedia, 4(2), p. 1883.*
23. *Tan, P. N., Steinbach, M., & Kumar, V. (2018). Introduction to Data Mining (2nd ed.). Pearson.*
24. *Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.*

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). Classification and Regression Trees. Wadsworth.