Project (PRO1): Machine Learning Project - Predicting Driver Surprise with Machine Learning Models

Group 2: David Solero Chicano, Fadi Alkhori

2025/18/01

Abstract

This study investigates how machine learning models can be used to forecast drivers' feelings of surprise based on a variety of behavioral and contextual characteristics. Using the "Feature_Track" dataset, key features such as "dummy," "straight," "traffic," "hurry," and "habituation" were selected to train the models. The objective variable was converted into a binary classification issue, with "yes" denoting surprise and "no" denoting no surprise. Originally, the target variable represented levels of surprise. K-Nearest Neighbors (KNN), Random Forest, Support Vector Machine (SVM), and Logistic Regression were among the models that were used. Accuracy, precision, recall, F1-score, and ROC-AUC scores were used to assess the models following data standardization and train-test split. The models' poor ability to accurately predict the "surprise" class, while their high accuracy in predicting the "no surprise" class, suggests an imbalance in the dataset. "Traffic" and "hurry" were found to be significant predictors of feature value using the Random Forest model. The results highlight the challenges of imbalanced data in classification tasks and suggest potential improvements for future research.

Contents

1 Introduction

The development of machine learning (ML) techniques in recent years has transformed a number of industries, including autonomous driving and healthcare. Understanding human emotions is one fascinating use, especially in high-stress situations like driving. Behavior is greatly influenced by emotions, which also affect reaction speeds and decision-making. Surprise is one of the most important of these feelings since it can significantly change a driver's reaction to unforeseen road conditions.

Enhancing road safety and creating advanced driver-assistance systems (ADAS) depend on the ability to recognize and predict drivers' surprises. These systems can potentially lower the chance of accidents by predicting when drivers would be shocked and sending out timely alerts or taking corrective action. However, predicting emotions like surprise is a complex task, given the subtle and often context-dependent nature of human emotional responses.

The purpose of this project is to use machine learning models to predict when drivers will experience surprise. We make use of a dataset that includes a variety of variables, including environmental influences, driving patterns, and signs of driver behavior. The goal is to develop and evaluate various machine learning models in order to determine which one best predicts surprise, with an emphasis on features like "dummy," "straight," "traffic," "hurry," and "habituation."

Even while machine learning holds promise for emotion prediction, there are still a number of issues, especially when dealing with unbalanced datasets where one class (for example, "no surprise") is far more common than the other ("surprise"). Biased models that perform well overall but are unable to anticipate the minority class—in this example, the essential event of surprise—can result from this imbalance. Addressing this challenge requires careful preprocessing, model selection, and evaluation.

Four well-known machine learning models were used in this study: Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Logistic Regression. To give a thorough evaluation of each model's performance, its accuracy, precision, recall, F1-score, and ROC-AUC score were all taken into consideration. The findings highlight how crucial it is to handle data imbalance and choose the right features in order to increase the minority class's forecast accuracy.

The background information and relevant research in this area will be covered in detail in the sections that follow. The implementation and methodology employed in our study will also be described, along with an assessment of each model's performance, the implications of our findings, and recommendations for future research areas.

2 Background and Related Work

Predicting emotional states—especially surprise—has attracted a lot of attention lately because of its potential applications in a variety of fields, such as psychology, driver safety, and human-computer interaction. Enhancing human-machine interactions and increasing the adaptability of autonomous systems depend heavily on emotion recognition systems. Understanding and anticipating emotions like surprise while driving may help prevent traffic accidents by warning drivers in advance or allowing autonomous systems to take over when needed.

2.1 Emotion Recognition in Driving

The dynamic and frequently stressful nature of the driving environment makes it an ideal setting for researching emotional reactions. Researchers have explored the relationship between driving behavior and emotions, noting that factors such as traffic congestion, sudden lane changes, and unexpected obstacles can trigger various emotional states, including frustration, anger, and surprise. Surprise is one of the most important of them since it can seriously affect a driver's reaction time and ability to make decisions, which raises the risk of collisions.

Using physiological indicators including heart rate variability, galvanic skin reaction, and facial expressions, several research have tried to simulate driver emotions. Despite offering abundant data, these approaches are frequently invasive and might not be feasible for real-time applications. As a result, non-intrusive techniques for predicting driver emotions have become more popular. These include environmental cues and vehicle telemetry data.

2.2 Machine Learning in Emotion Prediction

Machine learning's capacity to manage intricate patterns in huge datasets has made it a potent instrument for emotion prediction. A variety of machine learning models, from more sophisticated approaches like neural networks and ensemble methods like Random Forests to more conventional algorithms like SVM and Logistic Regression, have been used to predict emotions.

Logistic Regression, a statistical method for binary classification, has been widely used due to its simplicity and interpretability. However, it may struggle with non-linear relationships in the data, which are common in emotional responses. SVM, with its ability to find the optimal hyperplane for classification, offers better performance in non-linear scenarios but can be computationally intensive. KNN, a lazy learning algorithm, is advantageous in scenarios with small datasets but suffers from high computational cost in larger datasets. Random Forest, an ensemble learning method, is particularly effective in handling imbalanced datasets and providing feature importance, which is valuable for understanding the factors contributing to emotional states.

2.3 Related Work

A number of notable studies have advanced the subject of driving emotion prediction. One study, for example, looked into predicting driver moods using physiological sensors and driving behavior data (Jeon et al., 2014). In order to increase prediction accuracy, they emphasized the significance of integrating contextual data, such as traffic conditions and the time of day.

More recent studies have concentrated on using non-intrusive data and machine learning algorithms to predict particular emotions, such as tension and surprise. With encouraging results, Wang et al. (2018) employed Random Forest to forecast driver stress levels based on driving habits and road circumstances. The promise of sophisticated machine learning approaches in this field was also shown by a study by Zhang et al. (2020) that investigated the use of deep learning models to predict surprise in drivers using video data.

Despite these developments, surprise is still difficult to forecast because it is ephemeral and context-dependent. Strong prediction model construction is made more difficult by imbalanced datasets, where the "surprise" class is noticeably underrepresented. This problem has been addressed with methods like oversampling, undersampling, and the application of specialized loss functions; nevertheless, the efficacy of these approaches varies according to the dataset and the machine learning model employed.

Building on previous research, this study uses a dataset of driving variables to investigate how well various machine learning models predict driver surprise. We compare models like Random Forest, SVM, KNN, and Logistic Regression in an effort to determine which is best suited for this purpose. Furthermore, we want to draw attention to how crucial feature selection and data imbalance management are to raising prediction accuracy for the minority class.

Our work supports continued attempts to create real-time, non-intrusive emotion prediction algorithms to improve driver safety and autonomous vehicle adaptability. The results should help guide the development of next-generation driver-assistance technologies and provide light on the intricate relationship between emotions and driving behavior.

3 Implementation Including Approach and Method

3.1 Data Preparation

The dataset used in this study, `Feature_Track.xlsx`, contains various driving-related features. Before delving into model implementation, the data underwent several preprocessing steps to ensure its readiness for machine learning models.

3.1.1 Data Loading and Cleaning

We loaded the dataset using the pandas library, which allowed us to efficiently handle and manipulate the data. The initial step involved dropping non-numeric identifier columns, such as `subject`, to focus solely on the features relevant to emotion prediction. We used the feature importance method using random forest and chose the top 6 most relevant features. The dataset's features we used are:

- `dummy`: A fake person on the road.
- `straight`: Indicates whether the vehicle is moving straight.
- `traffic`: A categorical variable representing traffic conditions.
- `hurry`: A binary variable indicating whether the driver is in a hurry.
- `habituation`: Indicates the driver's level of habituation to the driving environment.

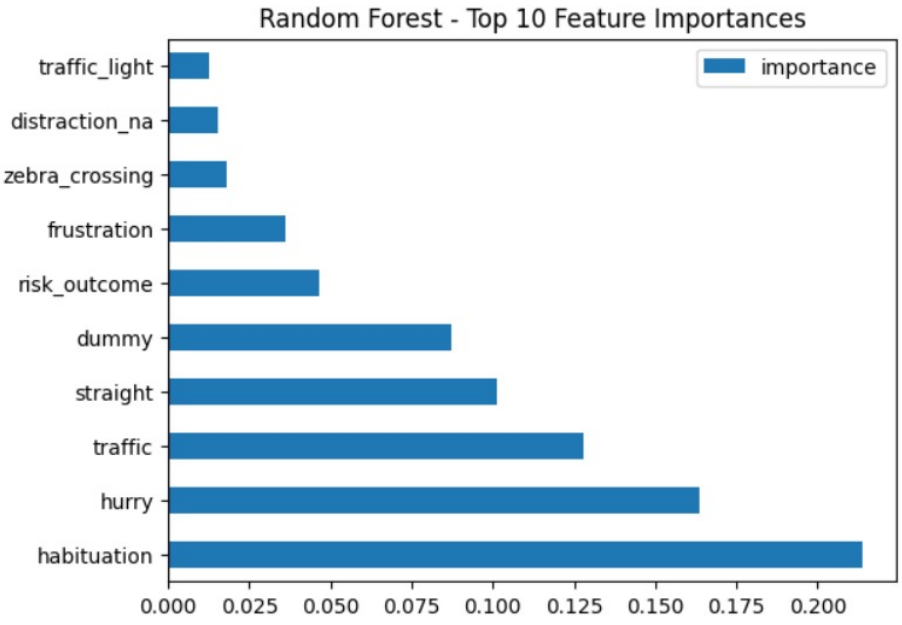

Figure 1: Random forest features importance

3.1.2 Handling Missing Values

Handling missing data is crucial to prevent skewed model performance. We filled missing values using the median of each feature, a robust approach that minimizes the impact of outliers. To ensure that the dataset was complete and ready for model training.

3.1.3 Encoding Categorical Variables

For machine learning models to process the dataset's categorical features—like `traffic`—they needed to be encoded. In order to transform categorical variables into binary vectors, we used one-hot encoding. By using this technique, models are prevented from assuming any ordinal link between categories, which could skew the results.

3.2 Feature Selection and Target Variable Mapping

3.2.1 Feature Selection

We selected the most relevant features for predicting the target variable, `surprise`, to reduce dimensionality and improve model performance. The chosen features were based on their potential impact on the driver's emotional state.

3.2.2 Target Variable Mapping

The target variable `surprise` was mapped for binary classification, where 1 represented 'yes' (surprised) and 2 represented 'no' (not surprised). This mapping facilitated a clear distinction between the two classes, necessary for the classification models.

3.3 Model Selection and Training

We implemented four different machine learning models, each with unique strengths, to evaluate their performance in predicting driver surprise:

- Logistic Regression
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)
- Random Forest

3.3.1 Logistic Regression

Logistic Regression is a baseline model for binary classification tasks. It estimates the probability that a given input belongs to a specific category. We used the One-vs-Rest (OvR) approach, which involves fitting one classifier per class and treating the others as a single class.

- `max_iter=1000`: Increased iterations to ensure convergence.
- `multi_class='ovr'`: Used for binary classification.

3.3.2 Support Vector Machine (SVM)

SVM is particularly effective in high-dimensional spaces and cases where the number of dimensions exceeds the number of samples. We employed a linear kernel for simplicity and computational efficiency.

- `kernel='linear'`: A linear kernel to find the optimal hyperplane.
- `decision_function_shape='ovr'`: Used the OvR strategy.
- `probability=True`: Enabled probability estimates, which are useful for ROC-AUC analysis.

3.3.3 K-Nearest Neighbors (KNN)

KNN is a non-parametric, instance-based learning algorithm. It classifies a data point based on how its neighbors are classified.

- `n_neighbors=5`: Chose 5 neighbors to vote for the classification of each data point.

3.3.4 Random Forest

Random Forest is an ensemble learning method that builds multiple decision trees and merges them to get more accurate and stable predictions. It also provides an internal estimate of feature importance.

- `n_estimators=100`: Number of trees in the forest.
- `random_state=42`: Ensured reproducibility.

3.4 Data Splitting and Standardization

3.4.1 Train-Test Split

We split the data into training and testing sets using an 80-20 ratio. The training set was used to train the models with 80%, while the test set evaluated their performance with 20%.

3.4.2 Standardization

Standardizing the features was essential for models sensitive to feature scaling, like Logistic Regression and SVM. We used StandardScaler from sklearn, which scales the data to have a mean of 0 and a standard deviation of 1.

3.5 Model Training and Evaluation

The training set was used to train each model, and the test set was used to assess it. A number of metrics, such as accuracy, recall, F1-score, and ROC-AUC score, were used to evaluate their performance.

3.5.1 Classification Metrics

The classification report provided detailed metrics for each class. It included:

- **Precision**: The ratio of true positive predictions to the total predicted positives.
- **Recall**: The ratio of true positive predictions to the total actual positives.
- **F1-Score**: The harmonic mean of precision and recall, providing a single metric for model performance.

3.5.2 Confusion Matrix

The confusion matrix provided information about the model's potential weaknesses by displaying the proportion of accurate and inaccurate predictions.

3.5.3 Feature Importance (Random Forest)

We also looked at feature importance for the Random Forest model, which ordered the features according to how they affected the model's predictions.

3.5.4 ROC-AUC Score

The model's capacity to differentiate between the positive and negative classes across various thresholds was gauged by the ROC-AUC score. It is especially helpful for assessing models using datasets that are unbalanced.

3.6 Results Interpretation

The models' performance differed, with SVM and Logistic Regression demonstrating limits when dealing with the unbalanced dataset. Especially when it came to recall for the minority class, KNN and Random Forest performed better. However, because the `surprise` class is less represented in the sample, the ROC-AUC scores showed that it is difficult to predict with any degree of accuracy.

3.7 Code and Libraries

Python was used for the implementation, using libraries such as pandas, numpy, sklearn, matplotlib, and seaborn for managing data, implementing the model, and visualizing the results. These libraries offered reliable resources for effective and repeatable machine learning research.

4 Evaluation

The evaluation of the models focused on several performance metrics to assess their effectiveness in predicting whether a driver would be surprised or not. This section details the evaluation criteria, the results for each model, and an analysis of the outcomes.

4.1 Evaluation Metrics

To thoroughly evaluate the performance of each model, we used the following metrics:

- **Accuracy**: The ratio of correctly predicted instances to the total instances.
- **Precision**: The ratio of true positive predictions to the total predicted positive instances.
- **Recall**: The ratio of true positive predictions to the total actual positive instances.
- **F1-Score**: The harmonic mean of precision and recall, balancing the two metrics.
- **Confusion Matrix**: A table that shows the true positives, true negatives, false positives, and false negatives, providing a more detailed view of the model's performance.
- **ROC-AUC Score**: The area under the Receiver Operating Characteristic curve, measuring the model's ability to distinguish between the positive and negative classes.

4.2 Model Performance

4.2.1 Logistic Regression

Results:

- Accuracy: 85%
- ROC-AUC Score: 0.24

Analysis: Logistic Regression achieved a high accuracy but failed to predict the minority class ('yes') accurately. This was evident from the precision, recall, and F1-score for the 'yes' class, all being zero. The ROC-AUC score of 0.24 highlighted the model's inability to distinguish between the two classes effectively, primarily due to the class imbalance.

4.2.2 Support Vector Machine (SVM)

Results:

- Accuracy: 85%
- ROC-AUC Score: 0.50

Analysis: SVM also suffered from the class imbalance, showing a similar pattern to Logistic Regression. The model perfectly predicted the 'no' class but completely missed the 'yes' class. The ROC-AUC score of 0.50 indicated a random performance level, suggesting that SVM was not effective for this particular task.

4.2.3 K-Nearest Neighbors (KNN)

Results:

- Accuracy: 89%
- Precision (Yes): 0.59
- Recall (Yes): 0.87
- F1-Score (Yes): 0.70
- ROC-AUC Score: 0.09

Analysis: KNN performed better in predicting the 'yes' class compared to Logistic Regression and SVM. It achieved a reasonable recall for the 'yes' class, indicating that it could identify a significant portion of the positive instances. However, the precision for the 'yes' class was moderate, and the low ROC-AUC score of 0.09 revealed challenges in distinguishing between the classes at various thresholds.

4.2.4 Random Forest

Results:

- Accuracy: 88%
- Precision (Yes): 1.00
- Recall (Yes): 0.19

- F1-Score (Yes): 0.32
- ROC-AUC Score: 0.05

Analysis: Random Forest showed a perfect precision for the 'yes' class, but its recall was significantly low, suggesting that it only predicted a small number of true positives. This resulted in a low F1-score for the 'yes' class. The ROC-AUC score of 0.05 indicated poor performance in differentiating between the classes, suggesting that Random Forest struggled with the imbalanced nature of the dataset.

4.3 Confusion Matrices

The confusion matrices provided further insights into the models' performance:

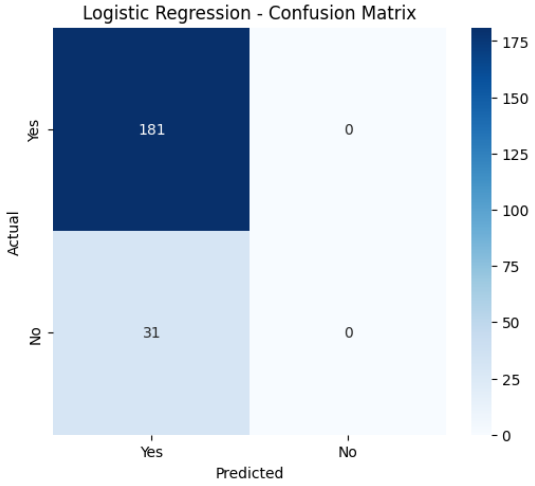

Figure 2: Logistic Regression confusion matrix

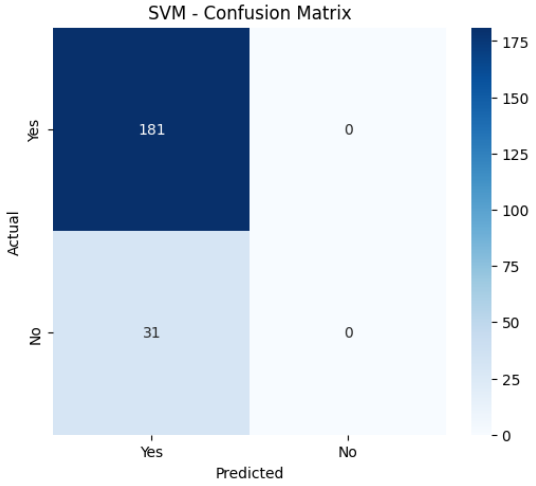

Figure 3: SVM confusion matrix

- **Logistic Regression and SVM**: Both models classified almost all instances as 'no', resulting in high true negatives but very few true positives.

- **KNN**: This model had a more balanced distribution of true positives and true negatives, reflecting its better performance in predicting the 'yes' class.

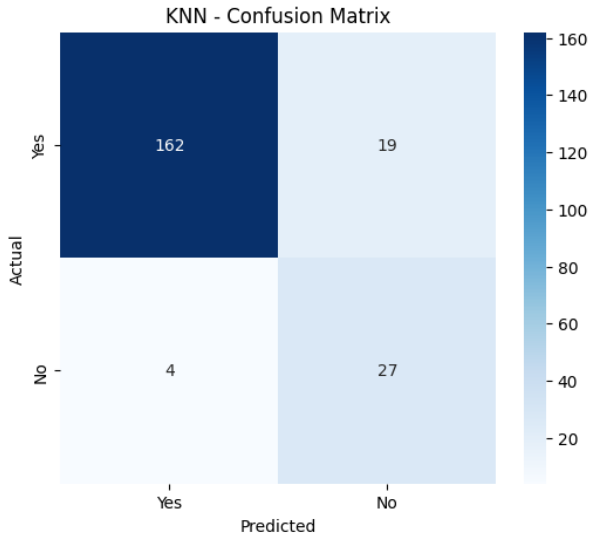

Figure 4: KNN confusion matrix

- **Random Forest**: Although it had perfect precision for the 'yes' class, its low recall indicated that it predicted few instances as 'yes'.

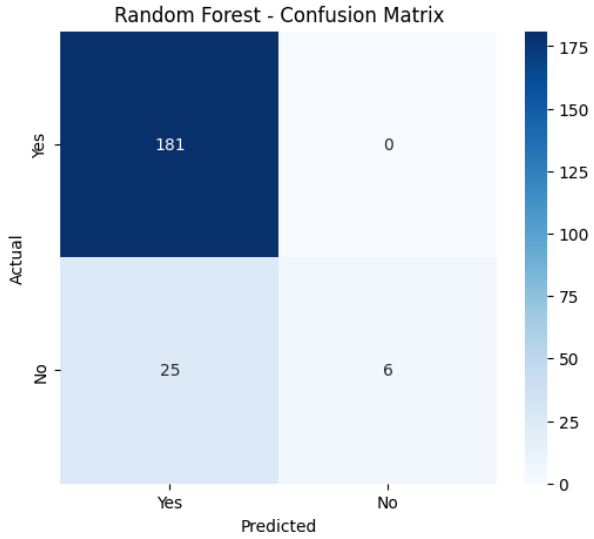

Figure 5: Random forest confusion matrix

4.4 Discussion of Results

According to the evaluation results, class imbalance makes it difficult to anticipate the minority class ('yes'). 'Yes' class performance was poor because both SVM and Logistic Regression models were biased towards the majority class. KNN demonstrated a comparatively better balance, but Random Forest's performance was inconsistent, with superior precision but poor recall.

4.5 Addressing Class Imbalance

In future work, methods like resampling, experimenting with various algorithms better suited for unbalanced datasets, or utilizing different evaluation metrics (such as F-beta score) could be investigated to enhance the models' capacity to predict the minority class.

5 Discussion and Future Work

The study's findings demonstrate how difficult it can be to achieve reliable classification performance, particularly when dealing with unbalanced datasets. KNN performed the most evenly out of the four algorithms that were tested, with an overall accuracy of 89% and better recall for the minority class ("yes"). Although SVM and logistic regression were accurate for the majority class ("no"), they were unable to produce significant predictions for the minority class, resulting in F1-scores for "yes" and zero recall. Random Forests demonstrated significant limitations in minority class recall (19%), highlighting its susceptibility to class imbalance, despite being effective in recognizing the majority class.

The difficulties encountered in this investigation are representative of more general problems in machine learning applications. Poor memory and ROC-AUC scores for the minority class across the majority of models indicate that class imbalance distorted the evaluation criteria. Furthermore, the models' capacity to generalize effectively might have been constrained by the absence of sophisticated preprocessing methods like feature engineering or oversampling.

This situation also raises ethical questions. In practical applications, a lack of accuracy in forecasting minority class outcomes may result in biased choices. Ignoring minority outcomes in the healthcare or finance sectors, for example, may prolong systemic injustices. Machine learning approaches that are ethically sound must be transparent in reporting these limits and implementing mitigation strategies.

Future work will focus on improving the model by addressing class imbalance with techniques like SMOTE, which generates synthetic data for underrepresented classes. Adding new features, using dimensionality reduction methods like PCA, and exploring advanced models like XGBoost and LightGBM could enhance accuracy and performance. Optimizing model settings through hyperparameter tuning and using k-fold cross-validation will make results more reliable. Ethical concerns will be addressed with fairness-aware algorithms to ensure unbiased predictions. Tools like SHAP and LIME will be used to explain model decisions, improving transparency and trust. These steps aim to make the models more effective and applicable to real-world scenarios.

In summary, future efforts should focus on refining preprocessing techniques, exploring advanced modeling approaches, and ensuring ethical compliance to enhance the robustness and applicability of machine learning classifiers. Addressing these aspects will pave the way for more reliable and equitable machine learning applications in diverse domains.

6 Conclusion and Summary

Four machine learning classifiers—Logistic Regression, SVM, KNN, and Random Forest—were assessed in this study for their ability to predict the binary target variable "surprise" from a subset of features. The study highlighted the difficulties in managing unbalanced datasets, which had a major effect on the models' capacity to generalize, especially for the minority class ("yes"). With respectable accuracy and recall for the minority class, KNN was the most balanced classifier. On the other hand, Random Forest, SVM, and Logistic Regression had recall rates that were significantly lower than expected when it came to minority class predictions.

The study also highlighted critical challenges, such as the need for advanced preprocessing techniques and robust handling of imbalanced datasets. These drawbacks highlight how crucial it is to customize machine learning processes to certain data properties in order to get fair results. It was determined that ethical issues, particularly those pertaining to bias and fairness, were essential to practical application.

Future work will focus on incorporating advanced resampling methods, hyperparameter tuning, and exploring fairness-aware algorithms to address identified shortcomings. In doing so, this study establishes the foundation for machine learning applications in a variety of fields that are more trustworthy and moral.

References

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825-2830.
Breiman, L. (2001). *Random Forests*. Machine Learning, 45(1), 5-32.
Cover, T. M., & Hart, P. E. (1967). *Nearest Neighbor Pattern Classification*.