# Random Forest Classification Tutorial

**Module Name:** Machine Learning and Neural Networks

**Submitted by:** Yasaswini Sure

**Student ID:** 24081945

**Tutor:** Peter Scicluna

**Assignment Type:** Individual Report

**Submission Date:** 11 Dec 2025

**GitHub link:** https://github.com/yashumcu0304/Random-Forest-Classification_Tutorial

# Random Forest Classification Tutorial

## Table of Contents

# Random Forest Classification Tutorial

## 1. Introduction

The Adult Census Income dataset is a widely used benchmark for evaluating supervised machine learning algorithms, particularly in binary classification tasks involving socioeconomic predictors. The objective is to predict whether an individual earns **more than $50,000 per year**, based on demographic and employment-related attributes. The dataset contains both categorical and numerical variables, missing values, and moderate class imbalance, making it an appropriate case study for demonstrating a **complete end-to-end machine learning workflow**.

This tutorial applies the **Random Forest classifier**, an ensemble method built on decision trees, to model income prediction. The workflow includes: data preprocessing, baseline modelling, hyperparameter tuning, model comparison, interpretability through permutation importance, and fairness considerations. The goal is not only to train a high-performing model but also to explain *why* it behaves the way it does, offering MSc Data Science students an accessible yet rigorous understanding of Random Forests.

Random Forests operate by training multiple decision trees and aggregating their predictions. To understand the ensemble, we begin with the structure of a single decision tree. Figure 1 (Section 2.2) illustrates how a tree splits the data based on impurity reduction until reaching terminal leaf nodes. Random Forests extend this by introducing randomness in sampling and feature selection, leading to diverse trees whose aggregated output improves generalisation. Figure 2 (Section 2.4) demonstrates majority voting, the mechanism through which the ensemble produces its final prediction.

This tutorial emphasises interpretability and fairness alongside performance, aligning with contemporary expectations in ethical AI practice.

## 2. Conceptual Foundations of Random Forests

Random Forests belong to the family of **ensemble learning algorithms**, which combine multiple base models to reduce variance and improve predictive power. In classification settings, each base learner is a decision tree trained on a modified version of the dataset.

### 2.1 Decision Trees as Base Learners

A decision tree recursively partitions the feature space by choosing splits that maximise class separation. At each internal node, the algorithm selects the feature and threshold that minimise impurity, typically measured using the **Gini index**:

$$\boldsymbol{G = 1 - \sum_{k=1}^{k} \rho_k^2}$$

where $\boldsymbol{p_k}$ the proportion of class $\boldsymbol{k}$ within the node. Trees grow until reaching stopping criteria such as maximum depth or minimum samples per leaf. While intuitive, individual trees tend to overfit, motivating the use of ensembles.

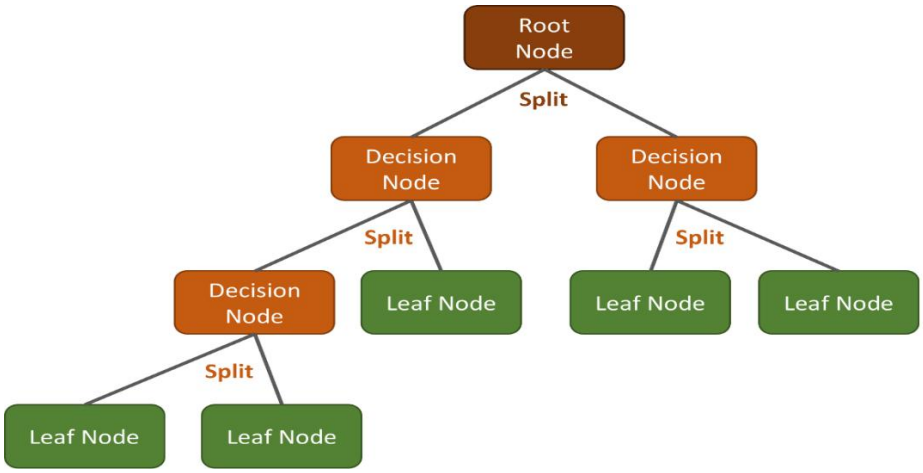
### 2.2 Tree Structure Explanation

Figure 1 shows the structure of a decision tree, including:

- **Root node**: initial split based on the most informative feature.
- **Internal nodes**: decision points partitioning the data.

- **Leaf nodes**: class predictions based on majority labels.

This structure allows trees to model non-linear decision boundaries, though at the cost of high variance when used alone.




### 2.3 Bagging and Feature Subsampling

Random Forests incorporate randomness through **bootstrap aggregation (bagging)**. Each tree is trained on a bootstrapped sample (sampling with replacement) of the training data. Additionally, only a random subset of features is considered at each split. These two mechanisms ensure that trees are diverse and weakly correlated, improving model robustness.

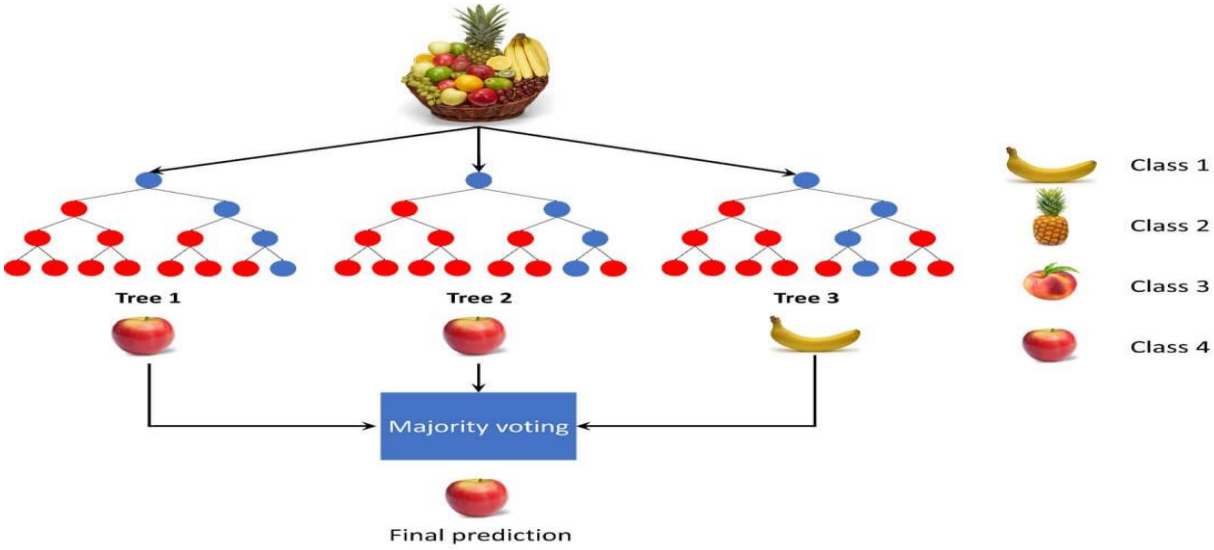
### 2.4 Majority Voting Mechanism

For classification, each tree outputs a predicted class. The Random Forest aggregates these predictions using **majority voting**:

$$\hat{y} = \boldsymbol{mode}(\boldsymbol{h_1}(\boldsymbol{x}), \boldsymbol{h_2}(\boldsymbol{x}), \ldots, \boldsymbol{h_T}(\boldsymbol{x}))$$

where $\boldsymbol{h_t}(\boldsymbol{x})$ is the prediction of tree **t**, and **T** is the number of trees.

Figure 2 illustrates majority voting with a conceptual example, showing how aggregated predictions outperform individual trees.



# Random Forest Classification Tutorial

### 2.5 Bias Variance Trade-off

A single deep tree has low bias but high variance. A Random Forest reduces variance by averaging over many de-correlated trees while maintaining relatively low bias, achieving an improved bias–variance balance suitable for structured tabular data.

### 2.6 Strengths and Limitations

**Strengths**:

- Handles mixed data types.
- High accuracy and robustness
- Resistant to overfitting
- Built-in feature importance
- Requires minimal feature engineering.

**Limitations**:

- Limited interpretability relative to single trees
- Biased impurity-based feature importance
- Computationally heavier than simple models
- May obscure fairness issues unless explicitly analysed.

## 3. Dataset Description and Preprocessing

### 3.1 Dataset Overview

The Adult Census Income dataset contains records from the 1994 US Census. Each observation includes demographic, educational, and employment details. The target variable, income, has two categories: <=50K and >50K.

### 3.2 Features and Target

Categorical features include marital.status, occupation, and workclass, while numerical features include age, hours.per.week, and capital.gain. The target variable was encoded as:

- 0 → <=50K
- 1 → >50K

### 3.3 Handling Missing Data

Missing values are represented by the symbol "?". These were converted to proper missing entries and removed. Although Random Forests can handle missingness to some extent, removing ambiguous rows simplifies analysis and avoids unclear categorical meanings.

### 3.4 Encoding and Scaling

A **ColumnTransformer** was used to:

- One-hot encode categorical variables

# Random Forest Classification Tutorial

- Standardise numerical features

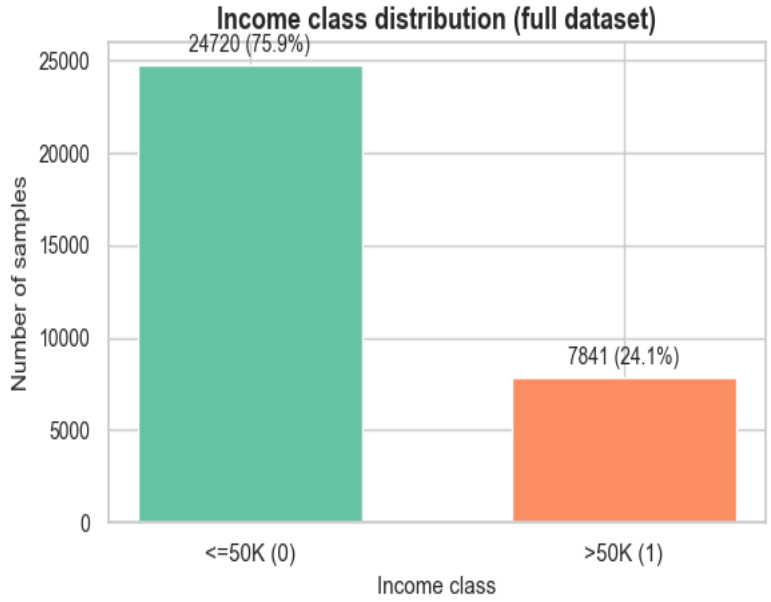
These transformations were embedded in a scikit-learn **Pipeline**, ensuring consistent preprocessing across training and test sets and preventing data leakage.

### 3.5 Train–Test Split and Class Balance

A stratified 80-20 split preserved the original class distribution. As Figure 3 shows, approximately **76%** of entries belong to the <=50K class, reflecting moderate imbalance.

### 3.6 Class Balance Visualisation

Figure 3 visualises the distribution of the two classes. Understanding this imbalance is essential for interpreting the precision and recall of the minority class.




## 4. Baseline Random Forest Model

### 4.1 Model Setup

A baseline Random Forest with near default hyperparameters was trained using the preprocessing pipeline. This establishes a performance benchmark before hyperparameter optimisation.

### 4.2 Training and Evaluation

The baseline model was evaluated on the test set using accuracy, precision, recall, F1-score, and ROC AUC, providing a multidimensional view of performance under imbalance.

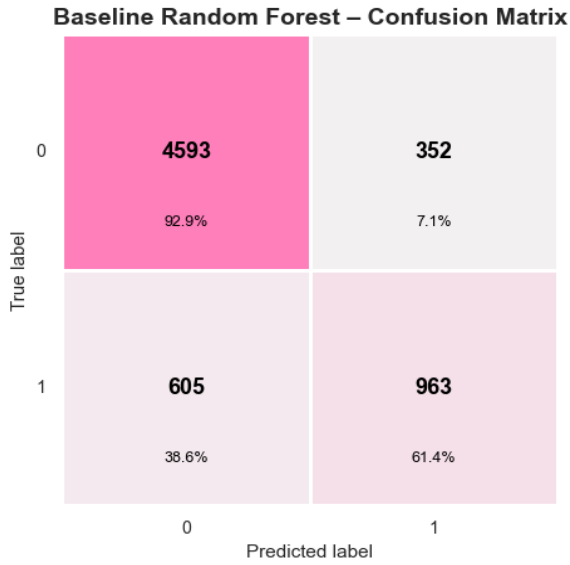
### 4.3 Baseline Metrics

The model achieved high accuracy but lower recall for the >50K category, indicating difficulty identifying high-income individuals.

### 4.4 Baseline Confusion Matrix

Figure 4 summarises predictions for both classes.

# Random Forest Classification Tutorial




### 4.5 Interpretation

Key observations:

1. Strong performance for the majority class
2. Moderate under-detection of high-income individuals
3. Motivation for improving minority-class recall through tuning.

## 5. Hyperparameter Tuning

### 5.1 Motivation

By adjusting tree depth, number of trees, and sampling parameters, the Random Forest's sensitivity to minority classes can be improved.

### 5.2 RandomizedSearchCV Setup

A RandomizedSearchCV was configured to explore:

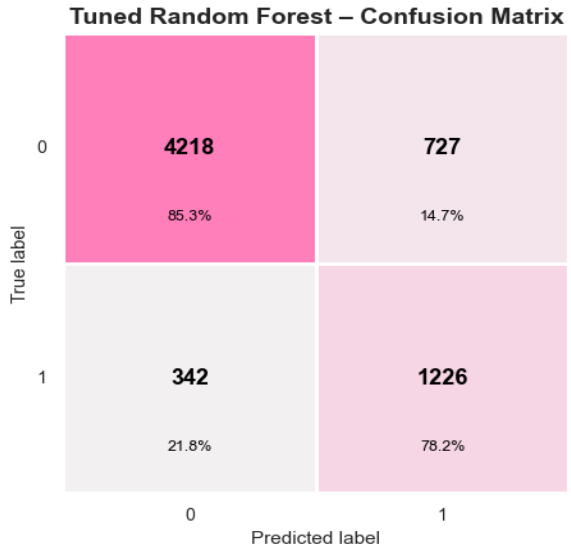
- n_estimators
- max_depth
- min_samples_split
- min_samples_leaf
- max_features

Cross-validation ensured robust performance estimation.

### 5.3 Tuned Model Performance

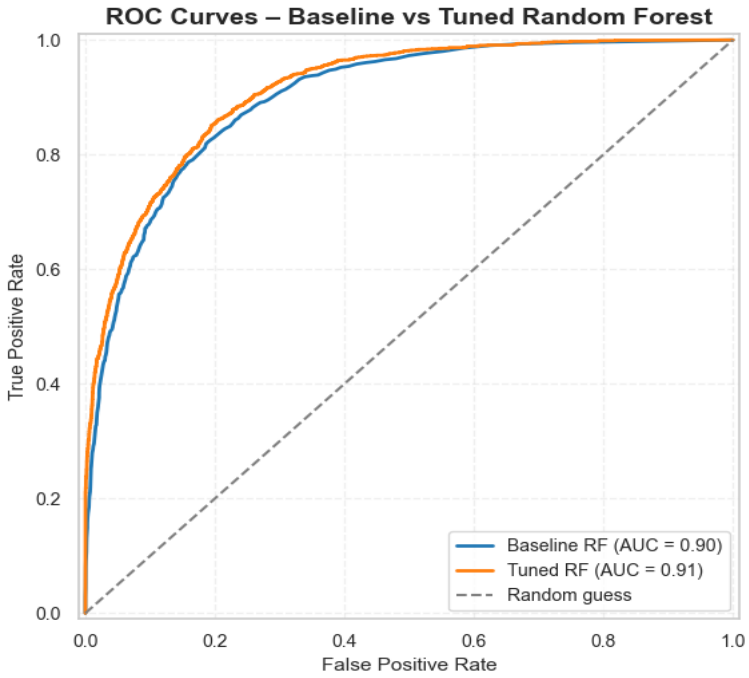
Figure 5 illustrates the tuned confusion matrix, showing significantly improved detection of the >50K class.

# Random Forest Classification Tutorial




### 5.4 ROC Curves

Figure 6 compares ROC curves for baseline and tuned models. The tuned model achieves a slightly higher AUC and consistently dominates across thresholds.

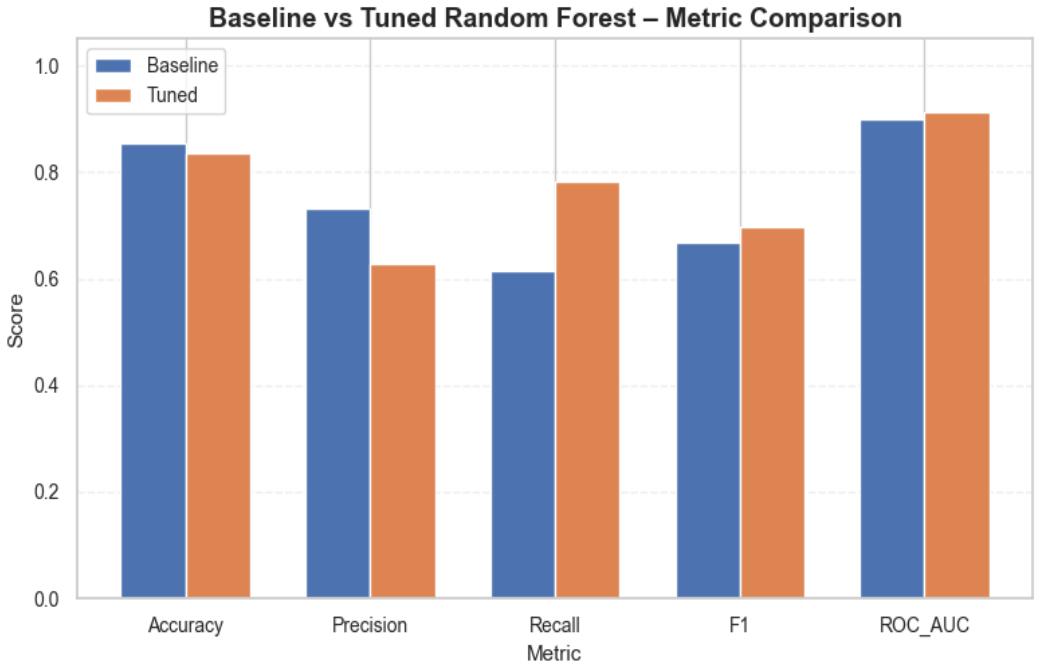



### 5.5 Metric Comparison

Figure 7 compares key evaluation metrics. The tuned model shows:

- Higher recall for the minority class
- Improved F1 score
- Slightly lower overall accuracy (expected under imbalance)

# Random Forest Classification Tutorial



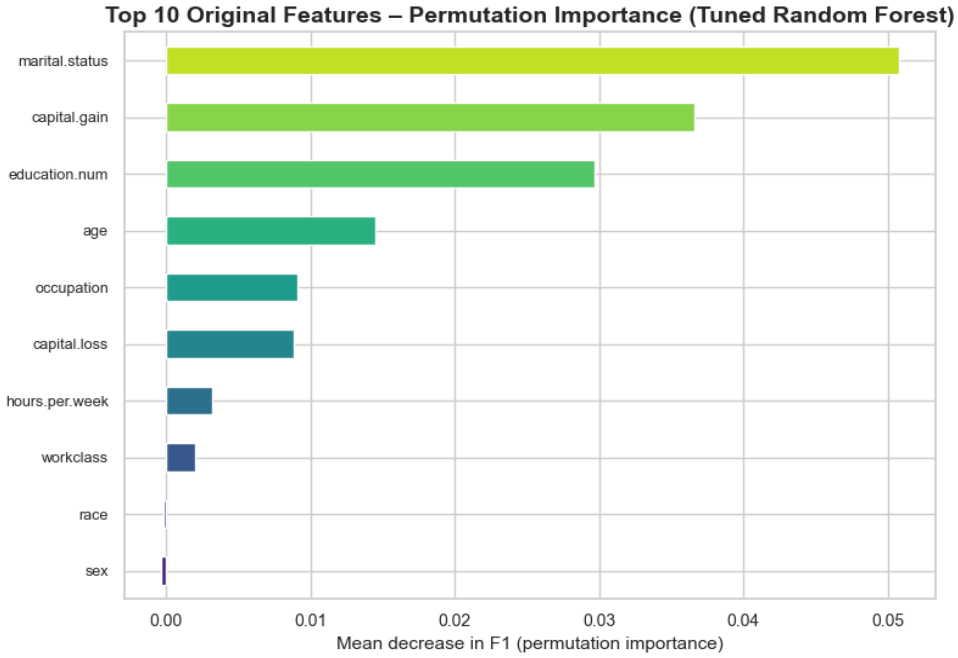

## 6. Interpretability and Feature Importance

### 6.1 Importance of Interpretability

Because socioeconomic data can influence decisions affecting individuals, interpretability is essential for transparency and fairness.

### 6.2 Permutation Feature Importance

Permutation importance measures the decrease in model performance caused by shuffling each feature. This provides a robust, model-agnostic measure of importance.



# Random Forest Classification Tutorial

### 6.3 Interpretation

Most important features include:

- marital.status
- capital.gain
- education.num
- age

These relate logically to income patterns. Sensitive attributes such as sex and race show minimal importance, suggesting limited reliance, though indirect bias remains possible.

## 7. Fairness and Ethical Considerations

### 7.1 Group-wise Evaluation

Metrics were compared across male and female subgroups. Performance was broadly similar, and sensitive features had low importance, indicating minimal direct bias.

### 7.2 Ethical Implications

Even small imbalances can propagate inequalities in real-world settings. Indirect bias from correlated variables remains a concern. Therefore, any operational deployment of such models requires monitoring, fairness audits, and transparent documentation.

## 8. Results and Discussion

### 8.1 Key Findings

The Random Forest models demonstrated strong predictive performance across the Adult Census Income dataset. The baseline classifier achieved high accuracy and reliably classified the majority <=50K group. However, it showed weaker sensitivity to the >50K class, reflecting the impact of class imbalance.

Hyperparameter tuning produced substantial improvements. The tuned model achieved markedly higher recall and F1-score for the minority class, showing that model sensitivity can be significantly enhanced through parameter optimisation. ROC AUC also increased, indicating better overall separability between income classes. These results collectively demonstrate the value of systematic tuning rather than relying solely on default settings.

### 8.2 Strengths

The final model exhibits several notable strengths:

- **Consistent performance on tabular data:** Random Forests are well suited for structured, mixed-type datasets and performed robustly across different evaluation metrics.
- **Meaningful interpretability:** Permutation importance revealed intuitive socioeconomic predictors such as marital status, capital gains, education level, and age.
- **Fairness indicators:** Minimal reliance on sensitive features (e.g., race, sex) suggests that the model does not directly encode demographic bias, although indirect effects remain possible.
- **Stability:** The ensemble approach reduces variance and makes the model less sensitive to data perturbations.

### 8.3 Weaknesses

Despite strong overall performance, several limitations were observed:

- **Impact of class imbalance:** The dataset's skew towards the <=50K class continues to affect the recall and precision of the minority class, even after tuning.
- **Interpretability challenges:** Although feature importance aids understanding, Random Forests remain less transparent than simpler models such as logistic regression or single decision trees.
- **Limited fairness exploration:** While basic group-wise comparisons were conducted, deeper fairness metrics (e.g., equalised odds) were beyond the scope of this tutorial.

These considerations highlight the importance of combining performance with responsible evaluation practices.

## 9. Limitations and Future Work

### 9.1 Dataset Limitations

Several characteristics of the Adult Census Income dataset impose constraints on modelling:

- **Self-reported variables:** Attributes such as occupation and education may contain noise or inaccuracies, potentially affecting model reliability.
- **Temporal relevance:** The dataset reflects socioeconomic patterns from the 1990s, meaning learned relationships may not generalise to modern labour markets.
- **Imbalanced target variable:** With approximately 76% of individuals earning <=50K, model performance can be biased toward the majority class despite tuning.

### 9.2 Model Limitations

Although Random Forests are powerful, they also present inherent limitations:

- **Reduced transparency:** Interactions across hundreds of trees are difficult to interpret fully, limiting their suitability for high-stakes decision-making.
- **Biased impurity-based importance:** Traditional Gini-based importance favours high-cardinality features; permutation importance helps but is more computationally expensive.
- **Lack of interaction analysis:** The model captures nonlinear patterns but does not explicitly reveal feature interactions without additional tools.

### 9.3 Future Extensions

Several avenues could enhance both performance and interpretability:

- **Gradient boosting frameworks** such as XGBoost, LightGBM, or CatBoost often outperform Random Forests on tabular datasets and support more advanced handling of imbalanced classes.
- **SHAP value analysis**, which provides local and global explanations, could deepen interpretability and highlight complex interactions.
- **Formal fairness metrics** (e.g., demographic parity, equalised odds) could provide a more comprehensive assessment of bias.
- **Bayesian optimisation** could replace RandomizedSearchCV for more efficient hyperparameter tuning.

# Random Forest Classification Tutorial

- **Additional feature engineering**, such as combining work-related and education-related variables, may reveal stronger predictive relationships.

## 10. Conclusion

This tutorial presented a complete and rigorous demonstration of the Random Forest algorithm applied to the Adult Census Income dataset. Beginning with foundational principles, the report outlined how decision trees form the basis of Random Forests and how ensemble methods address variance and improve generalisation. A structured preprocessing pipeline ensured clean and consistent treatment of mixed data types, enabling reliable downstream modelling.

The baseline model provided a strong starting point, but hyperparameter tuning significantly enhanced the model's ability to detect high-income individuals an essential improvement given the dataset's imbalance. Evaluation through confusion matrices, ROC curves, and metric comparisons demonstrated measurable benefits from tuning.

Interpretability was explored through permutation feature importance, revealing sensible and intuitive predictors aligned with socioeconomic expectations. Fairness considerations highlighted the importance of evaluating the influence of sensitive attributes and offered guidance for responsible deployment.

Overall, the tutorial illustrates how Random Forests can be effectively trained, tuned, and interpreted in a practical classification setting while emphasising methodological rigour, transparency, and ethical awareness. This workflow provides a transferable template that MSc Data Science students can apply to similar machine learning problems in both academic and professional contexts.

## 11.References:


- DataHacker.rs (n.d.) *Machine Learning: Introduction to Random Forest*. Available at: https://datahacker.rs/012-machine-learning-introduction-to-random-forest/ (Accessed: 2 November 2025).
- Freie Universität Berlin (n.d.) *Random Forest – RESEDA Research Group*. Available at: https://blogs.fu-berlin.de/reseda/random-forest/ (Accessed: 12 November 2025).
- Prashant (n.d.) *Random Forest Classifier Tutorial*. Kaggle. Available at: https://www.kaggle.com/code/prashant111/random-forest-classifier-tutorial (Accessed: 2 December 2025).
- Intellipaat (2020) *Random Forest Explained | Random Forest Algorithm in Machine Learning | Data Science*. YouTube. Available at: https://www.youtube.com/watch?v=vdfvQTi65og (Accessed: 25 November 2025).
- Simplilearn (2020) *Random Forest Algorithm – Random Forest Explained | Random Forest in Machine Learning*. YouTube. Available at: https://www.youtube.com/watch?v=eM4uJ6XGnSM (Accessed: 29 November 2025).
- StatQuest with Josh Starmer (2018) *Random Forests Part 1: Building, Using and Evaluating Random Forests*. YouTube. Available at: https://www.youtube.com/watch?v=J4Wdy0Wc_xQ&t=280s (Accessed: 4 December 2025).
- Kaggle (n.d.) *Adult Census Income*. Available at: https://www.kaggle.com/datasets/uciml/adult-census-income (Accessed: 1 November 2025).