Random Forest Model Analysis Report

1. Introduction

A **Random Forest** is a powerful machine learning model that combines multiple decision trees to make more accurate predictions. Instead of relying on a single tree, it builds many trees and averages their results, reducing errors and improving reliability. This report explains how different Random Forest settings performed and which works best for the given dataset.

2. Methodology

2.1 How Random Forest Works

- Builds **many decision trees** (called "estimators").
- Each tree trains on a **random subset of data** (bootstrapping).
- Splits nodes using a **random subset of features** (reduces overfitting).
- Final prediction = **average (regression) or majority vote (classification)** of all trees.

2.2 Steps Taken

Step 1: Data Preprocessing

- **Checked for missing values** (ensured clean data).
- **Exploratory Data Analysis (EDA):** Studied feature relationships.
- **Encoded categorical data** (converted text to numbers).
- **Split data:**
  - **70% Training set** (to train the model).
  - **30% Testing set** (to evaluate performance).

Step 2: Model Training

- Tested different **hyperparameters**:
  - n_estimators (number of trees).
  - max_depth (how deep trees can grow).
  - max_features (how many features to consider per split).
  - min_samples_split (minimum samples needed to split a node).

Step 3: Performance Evaluation

Measured using:

- **Accuracy** (how often predictions are correct).
- **Feature importance** (which features matter most).

3. Model Comparison & Results

Model Name	**Key Settings**	**Accuracy**	**Performance**
Default Configuration	No tuning	**0.9403**	**Best model** (works perfectly out of the box).
Increased Max Features	max_features="auto"	0.9161	Good but slightly worse than default.
Bootstrapping Disabled	bootstrap=False	0.8075	Much worse (bootstrapping is essential).
Increased Depth (max_depth=20)	Deep trees	0.8030	Overfitting risk (too complex).
More Trees (n_estimators=200)	Extra trees	0.7991	No improvement (default trees are enough).
Regularization (max_depth=10, min_samples_split=5)	Limits tree growth	0.7653	Too restrictive (hurts performance).

Key Findings:

1. **Default Random Forest works best (94.03% accuracy)** → No tuning needed.
2. **Increasing** max_features **helps slightly (91.61%)** but not better than default.
3. **Disabling bootstrapping or making trees too deep/complex reduces performance** → Simpler is better here.
4. **Adding more trees (**n_estimators**) doesn't help** → Default (100 trees) is sufficient.

4. Recommendations

Best Model Choice:

Use Default Random Forest (highest accuracy, no extra tuning needed).

If You Want to Experiment:

Try adjusting max_features (e.g., "auto", "sqrt") to see if accuracy improves.

Avoid These Settings:

- **Don't disable bootstrapping (**bootstrap=False**)** → Makes the model worse.
- **Don't increase depth (**max_depth=20**)** → Leads to overfitting.
- **Don't add unnecessary trees (**n_estimators=200**)** → Wastes time, no benefit.

General Tips:

✓ **Check feature importance** → Helps identify key predictors.
✓ **Use cross-validation** → Ensures stable results.
✓ **Stick with defaults first** → Only tune if needed.

5. Conclusion

- **Best Model: Default Random Forest (94.03% accuracy)** → Works perfectly without tuning.
- **Small tweaks to** max_features **may help**, but default is already excellent.
- **Avoid overcomplicating** (deep trees, extra estimators) → Hurts performance.

Final Decision: Use the **Default Random Forest** for the best results.