Contents

[Project Report 2](#_Toc170558557)

[Introduction 2](#_Toc170558558)

[Task 1 2](#_Toc170558559)

[Dataset Overview and Preprocessing 2](#_Toc170558560)

[Data Loading and Initial Exploration 2](#_Toc170558561)

[Preprocessing Steps: 2](#_Toc170558562)

[Handling Missing Data: 2](#_Toc170558563)

[Challenges: 2](#_Toc170558564)

[Methodology and Techniques: 3](#_Toc170558565)

[Results and Discussion: 3](#_Toc170558566)

[Model Comparison and Selection 5](#_Toc170558567)

[Hyperparameter Tuning result shows the best parameters for Random Forest model: 5](#_Toc170558568)

# Project Report

### Introduction

In this assignment, we demonstrate several machine learning techniques to solve classification problems. By processing the datasets (both numerical and categorical features), we explore the process of data preprocessing, model building, and evaluation. This assignment showed the importance of data preparation, model selection, and fine-tuning to achieve the most optimal performance in machine learning.

## Task 1

### Dataset Overview and Preprocessing

Dataset1 consisted of 925 rows and includes the features var1, var2, var3, var4, var5, var6, and var7, with the target column as the label. Among these features, var3 and var6 are categorical, while var7 contains datetime values. The other features such as var1, var2, var4, and var5, are numerical.

### Data Loading and Initial Exploration

The dataset was loaded and initial exploration was conducted to understand its structure and identify any data quality issues. The first step is checking for missing values, which revealed a significant amount of missing data in the var4 column. Missing Data in var4 about 600 missing values (about 65% missing).

### Preprocessing Steps:

For the categorical features, we converted var3 and var6 into numerical values using label encoding. Regarding the datetime feature, var7, we converted it to a datetime format and split it into var7\_date and var7\_time.

### Handling Missing Data:

Since var4 had around 65% missing values, we explored to impute missing values in a dataset using a KNN imputer. The KNN (K-Nearest Neighbours) imputer is a method that replaces missing values based on the mean or median values of the nearest neighbours. Since KNN works with distance computations, first we scale the dataset.

### Challenges:

We faced several challenges during the preprocessing phase of the dataset. One major issue was handling a significant amount of missing data in var4, which required careful consideration of imputation techniques or potential removal. Ensuring the correct encoding of categorical features, such as var3 and var6, was important to maintain the integrity of the data. Properly managing and splitting datetime values in var7 also posed a challenge, as it was need to accurately separate and format the date and time components for further analysis.

### Methodology and Techniques:

Model Selection: Applied logistic regression, decision trees, random forests, and artificial neural networks (ANN) to the classification problem.

Hyperparameter Tuning: Use Grid Search and cross-validation to fine-tune model parameters.

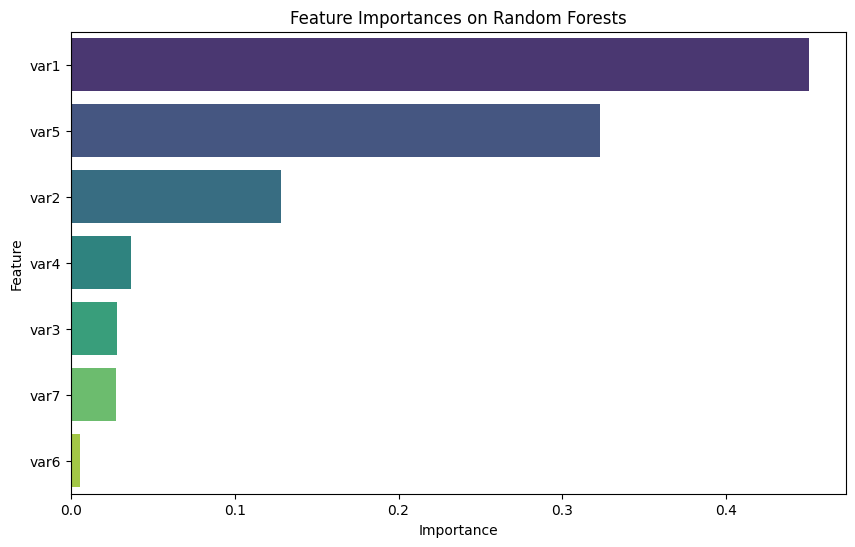
Evaluation Metrics: Assessed models using accuracy, precision, recall, and F1-score.

### Results and Discussion:

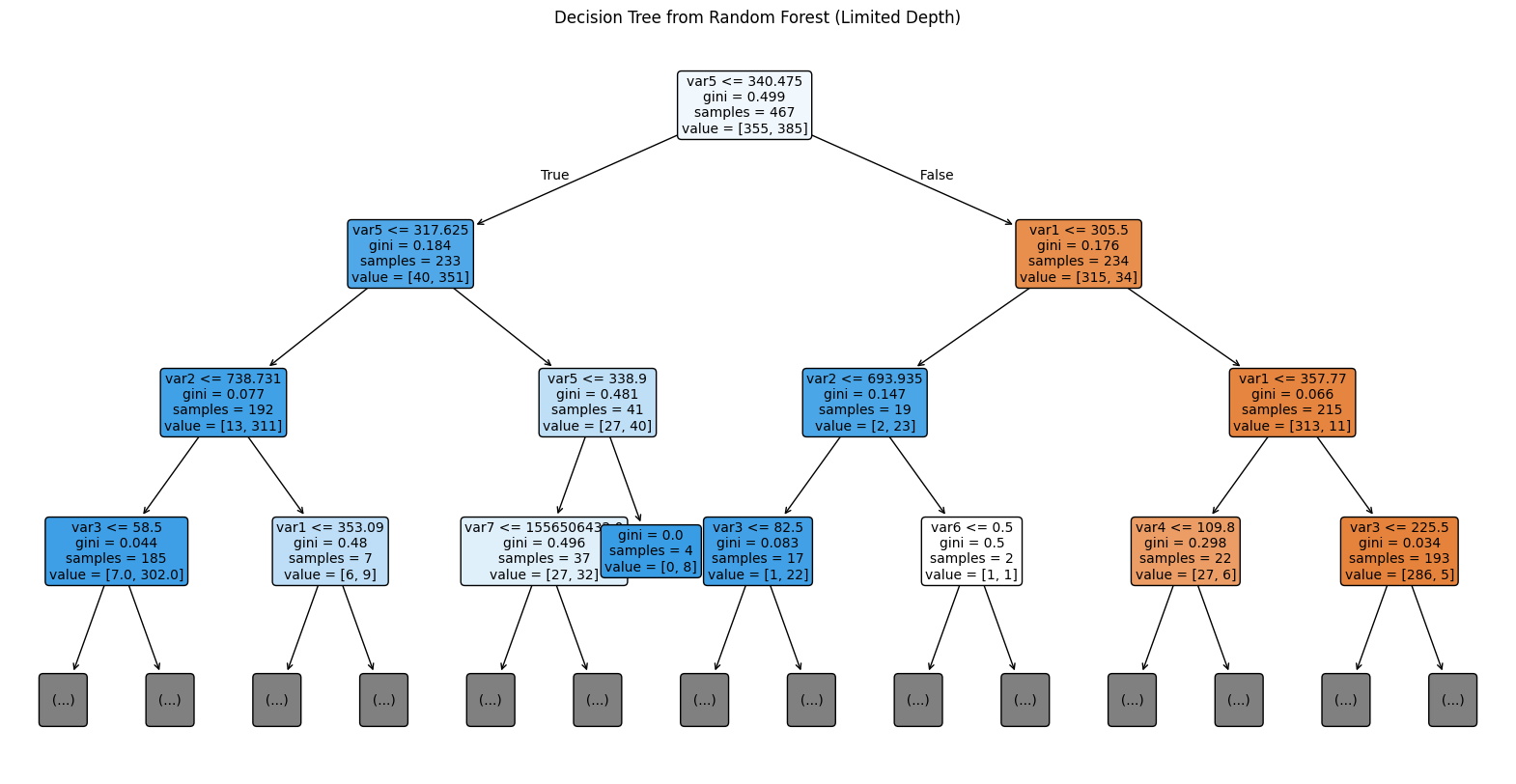
Random Forests model with or without normalization outperformed other models and achieved the highest accuracy, around 96.2%.

Important features for the Random Forest model are var1, var5, and var 2.

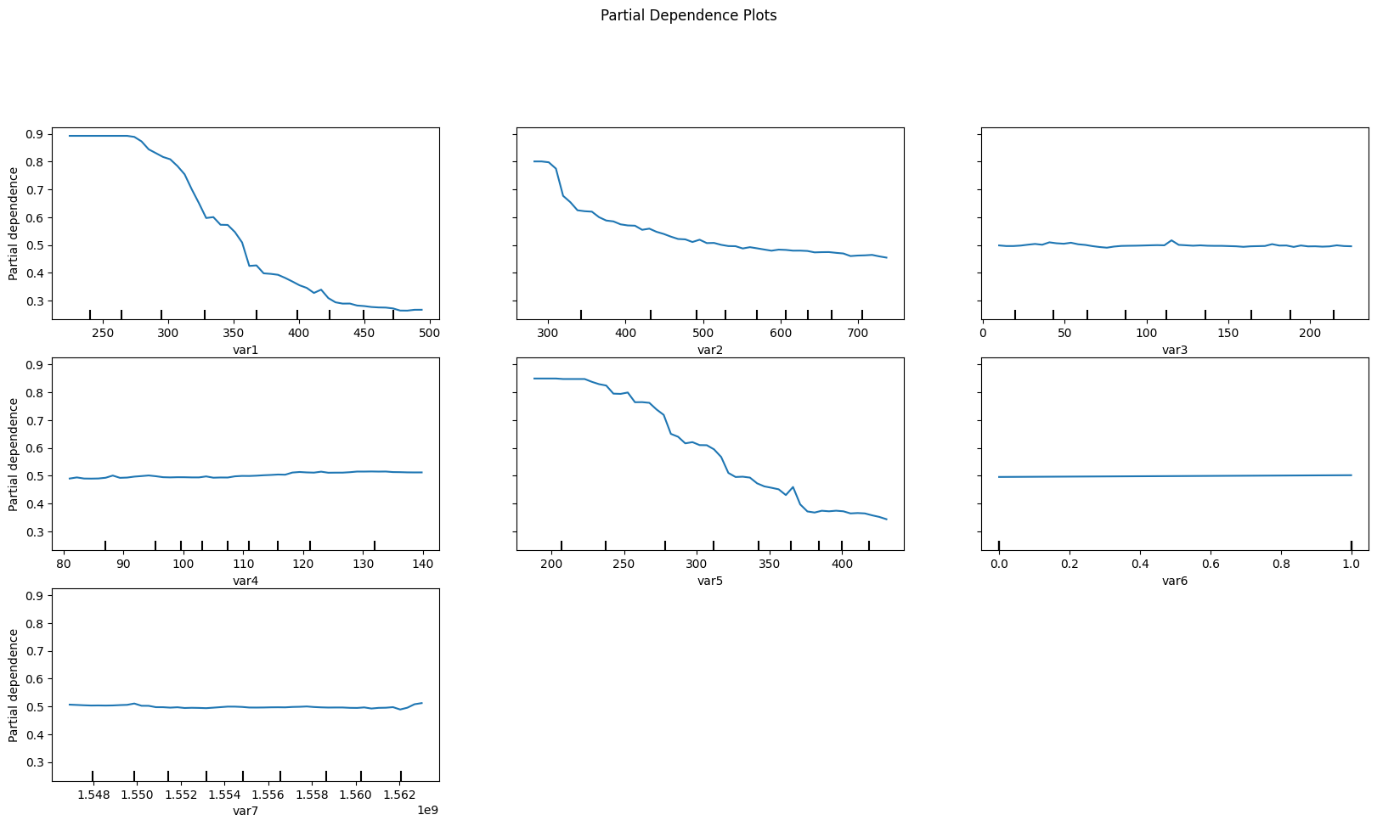
#### Feature importance



#### Limited Depth Decision Tree



#### Partial Dependence Plots



Since var4 is not a significant feature, imputing the mean or removing the column doesn’t affect the model's accuracy.

Confusion matrix and heatmap for the final selected model shows that the model has high and reliable performance.

### Model Comparison and Selection

Model Performance Comparison:

|  |  |
| --- | --- |
| Classifier | Accuracy |
| Random Forests | 0.9621621621621622 |
| Logistic Regression | 0.9513513513513514 |
| Decision Trees | 0.9405405405405406 |
| Artificial Neural Networks | 0.9621621370315552 |

### Hyperparameter Tuning result shows the best parameters for Random Forest model:

Best Parameters:

{

    'bootstrap': [False],

    'max\_depth': [10],

    'max\_features': ['sqrt'],

    'min\_samples\_leaf': [2],

    'min\_samples\_split': [2],

    'n\_estimators': [300]

}

**Conclusion**

Effective data preprocessing is important for model performance. Ensemble methods like Random Forest can provide significant improvements. Fine-tuning and cross-validation are necessary steps to optimize model performance.

In future, we can explore more advanced techniques for handling missing data and feature engineering. Experiment with other methods and deeper neural networks, and investigate the impact of feature selection and dimensionality reduction techniques.