

CCIA IT ML Prediction

CCIA.IT.50.SOP.0000

Version 1

Table of Contents

[1.1 Introduction 2](#_Toc10869155)

[1.2 Problem Overview 3](#_Toc316859234)

[1.3 Machine Learning Method: Random Forest 4](#_Toc1857031991)

[1.4 Dataset Overview 5](#_Toc1930317757)

[1.5 Methodology 6](#_Toc134844681)

[1.6 Explanation of the Code (Move-In Date) 6](#_Toc622131381)

[1.7 Explanation of the Code (DEST\_ACCOUNT) 12](#_Toc961124526)

[1.8 Result 15](#_Toc310970398)

## **Introduction**

In this project, we aim to build a machine learning model to predict the **move-in date** and the **destined account** for cattle based on their **birthdate** and **source account**. The data we use is provided in two CSV files: one containing **birthday information** for each cattle and another containing **move-in details**. By combining and processing these datasets, we will use a **Random Forest model**, a powerful machine learning algorithm, to make these predictions. The goal is to predict the following:

1. **Move-in Date:** Predicting when each cattle will move into its new location.
2. **Destined Account:** Determining the account that the cattle is assigned to based on its birthday and source account.

The dataset provided contains **27,378 observations** (records), and we will apply a **training/test split** (80/20) to evaluate the model's performance.

## **Problem Overview**

The move-in data consists of records with **tags** (unique identifiers for the cattle), along with their **move-in dates**. Each cattle also has a **birthday** recorded in a separate dataset. The challenge is to use the birthdate and source account information to predict the **move-in date** and the **destined account**.

To address this, we will:

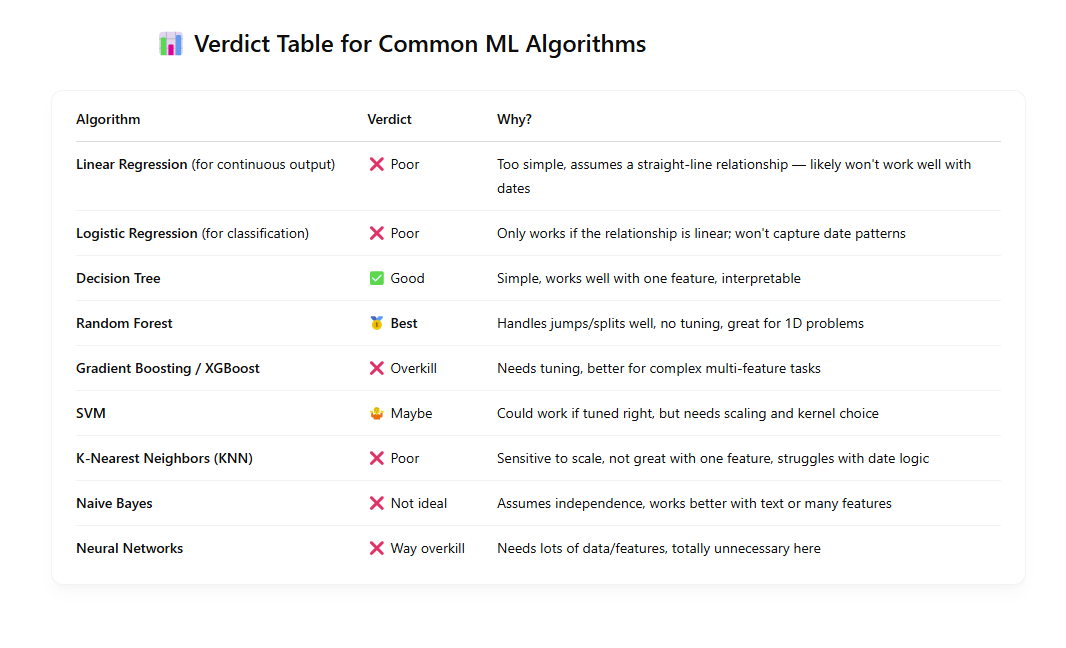
* Preprocess the data by cleaning and merging the two datasets.
* Perform **feature engineering** to generate new attributes that may improve the model's performance.
* Use **Random Forest**, a robust ensemble learning algorithm, to build the prediction model.

## **Machine Learning Method: Random Forest**

**Random Forest** is a popular machine learning algorithm that creates multiple decision trees and combines their outputs to improve accuracy and reduce overfitting. Random Forest works well with datasets that contain a mix of numerical and categorical data, making it a good choice for this project.

### **Why Random Forest?**

* **Non-linearity:** Random Forest can handle complex relationships between features, which might not be easy to capture with traditional linear models.
* **Robustness:** It is less prone to overfitting compared to other models like decision trees, especially with a large number of observations.
* **Feature Importance:** Random Forest can automatically assess the importance of each feature, which can help in understanding the underlying data.



## **Dataset Overview**

The dataset consists of the following features:

* **Birthdate**: The birthdate of each cattle.
* **Move-in Date**: The date when the cattle moves into its new location.
* **Source Account**: The account from which the cattle originated.
* **Destined Account**: The account to which the cattle is assigned.
* **TAG\_START**: Unique identifier for each cattle.

The goal is to predict:

1. **Move-in Date** (a date value) based on the birthdate and source account.
2. **Destined Account** (categorical value) based on the birthdate and source account.

The dataset contains **27,378 observations**, and we will split the data into **80% training data** and **20% test data** for model evaluation.

## **Methodology**

The data preprocessing steps involve:

* **Merging** the birthday and move-in datasets by the TAG\_START column (unique identifier).
* **Cleaning** the data by removing duplicates and irrelevant columns.
* **Feature Engineering**: Creating new columns such as the cattle’s age at move-in, the month and weekday of the birthdate and move-in date, etc.
* **Model Training**: Using the Random Forest algorithm to train a predictive model based on the processed data.

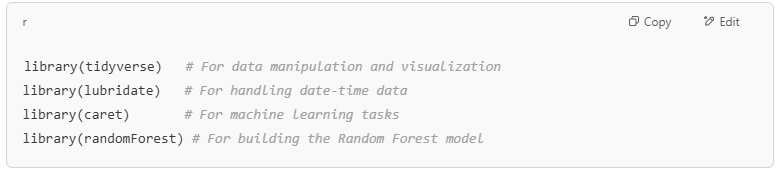
The following sections will explain each of the steps in detail, from data preparation and cleaning to building and evaluating the model.

## **Explanation of the Code (Move-In Date)**

This section builds a model to predict the **EVENT\_DATE** based on the **DOB** using a Random Forest classifier.

### **Loading Libraries**

To begin, we load the necessary libraries that will assist with data manipulation, machine learning, and model evaluation.



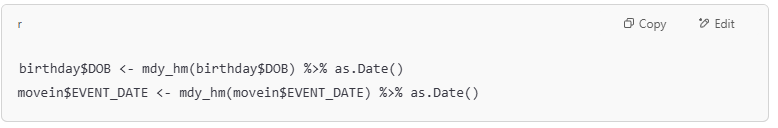
### **Reading the Dataset**

The dataset consists of two CSV files: Birthdate and MoveIn. We read them into R using the read.csv() function.



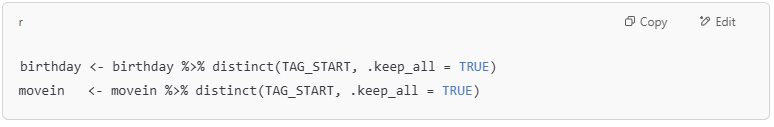
### **Date Conversion**

Since the original data includes timestamps along with the dates, we standardize the date format to yyyy-mm-dd to remove the time component.



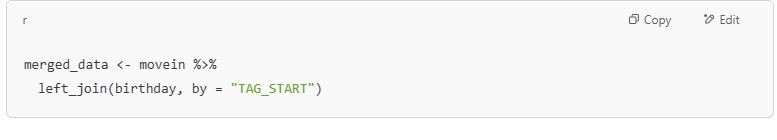
### **Removing Duplicate Records**

To ensure the integrity of the data, we remove any duplicate TAG\_START entries in both datasets.



### **Merging the Datasets**

We then merge the birthday data with the move-in data on the TAG\_START field to ensure each cattle has both their birthdate and move-in details in the same dataset.



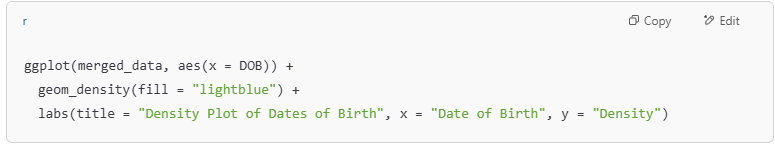
### Data Cleaning

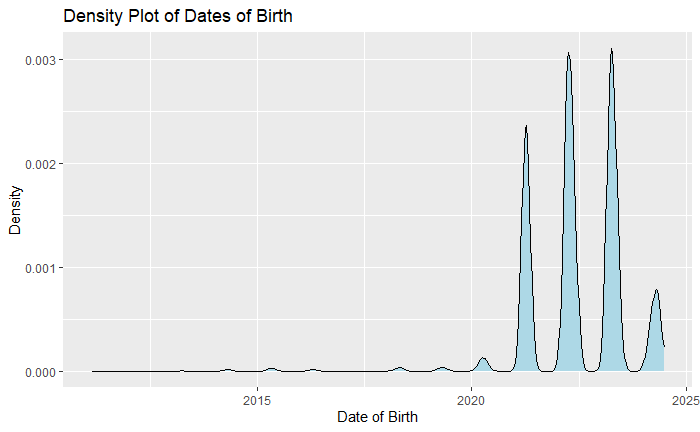
To begin with, we load and merge the birthday and move-in datasets using a common identifier. One of the columns, SOURCE\_PREMISES, contains mostly missing values and thus is removed from the dataset to reduce noise and avoid misleading the model.



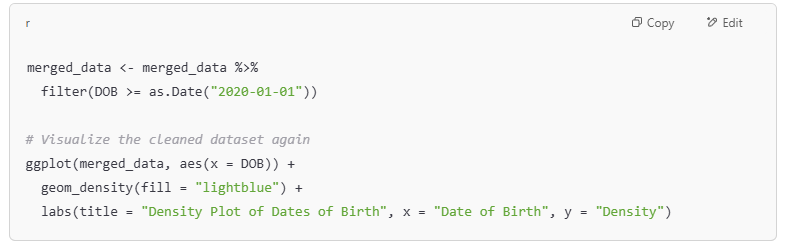
### Exploratory Data Analysis (EDA)

We visualize the distribution of the DOB (Date of Birth) column to better understand the spread of birth dates in our dataset.

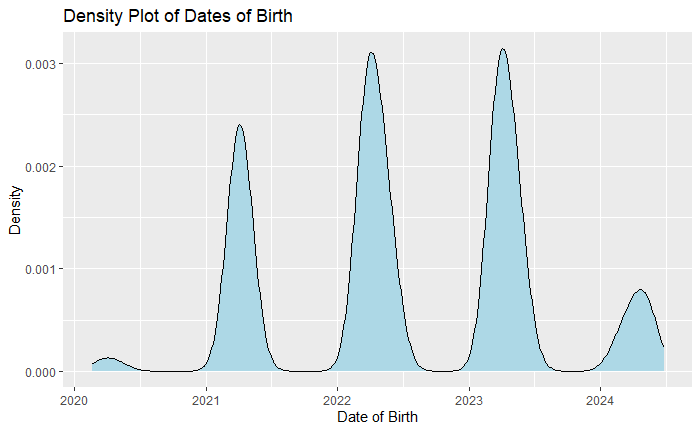




Upon inspection, some dates before the year 2020 are considered outliers and are filtered out to improve the reliability of our predictions.



After:



### Feature Engineering

To enhance the model's performance, we generate new features that can capture temporal patterns and relationships:

* age\_at\_movein\_days: Number of days between birth date and move-in date.
* dob\_month: Month of birth.
* dob\_weekday: Day of the week of birth.
* movein\_month: Month of move-in.
* movein\_weekday: Day of the week of move-in.

We also convert the EVENT\_DATE column to a numeric format by calculating the number of days since a fixed reference date (e.g., Jan 1, 2000), which simplifies prediction as a regression task.

### **Reminder: Apply Same Feature Engineering to New Data**

Before using your trained model to make predictions on new data, it's essential to **repeat the same feature engineering steps** you applied during training.

**Feature Engineering Steps to Repeat:**

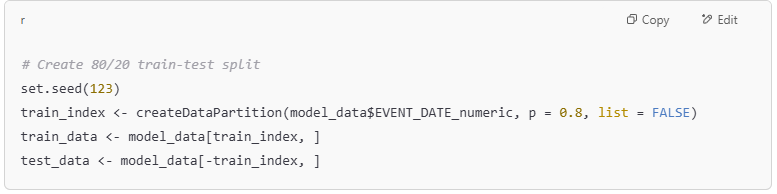
* age\_at\_movein\_days: Difference between EVENT\_DATE and DOB
* dob\_month: Extracted month from date of birth
* dob\_weekday: Day of the week for date of birth
* movein\_month: Extracted month from move-in date
* movein\_weekday: Day of the week for move-in date
* EVENT\_DATE\_numeric: Number of days since a reference date (e.g., 2000-01-01)

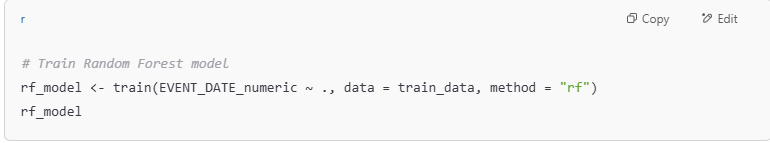
Also, **ensure categorical features (like weekdays or months) are converted to factors**, just like during training.



### Model Training (Random Forest)

We split the data into training (80%) and testing (20%) sets using createDataPartition from the caret package.

Then, we train a **Random Forest model** to predict the numeric representation of the move-in date using the engineered features.



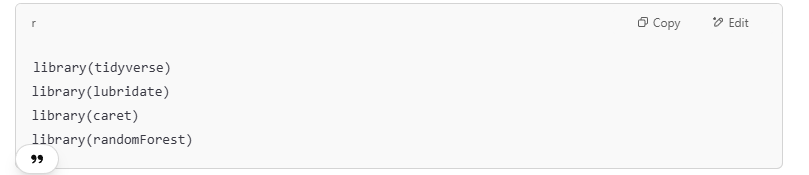
The output rf\_model is the trained Random Forest Model for Move-In Date

## **Explanation of the Code (DEST\_ACCOUNT)**

This section builds a model to predict the **DEST\_ACCOUNT** based on the **SOURCE\_ACCOUNT** using a Random Forest classifier.

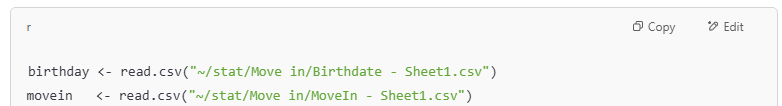
### **Loading Libraries**

To begin, we load the necessary libraries that will assist with data manipulation, machine learning, and model evaluation.



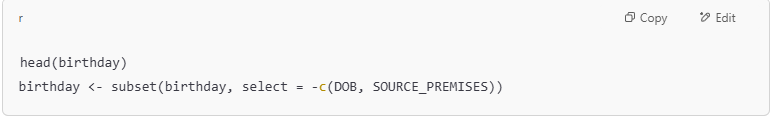
### **Reading the Dataset**

The dataset consists of two CSV files: Birthdate and MoveIn. We read them into R using the read.csv() function.



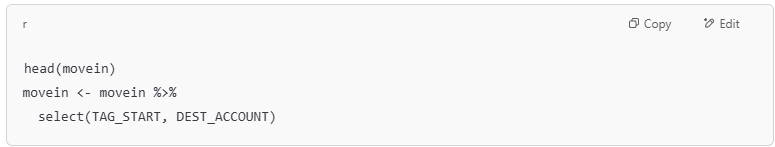
### Drop Unnecessary Columns

The DOB and SOURCE\_PREMISES columns are removed since DOB isn’t used here, and SOURCE\_PREMISES contains mostly missing values.



### Select Relevant Columns in Move-in Data

Only TAG\_START and DEST\_ACCOUNT are retained to simplify the dataset and focus on prediction.



### Merge Datasets and Check for Missing Values

The datasets are merged using the common key TAG\_START, and missing values are checked to ensure data quality.



### Convert Variables to Factors

SOURCE\_ACCOUNT and DEST\_ACCOUNT are converted to factors so they are treated as categorical variables during model training.



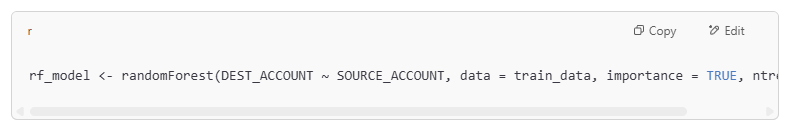
### Split Data into Training and Testing Sets

The data is split 80% for training and 20% for testing. Setting the seed ensures the split is reproducible.



### Train the Random Forest Model

A Random Forest classifier is trained to predict DEST\_ACCOUNT using SOURCE\_ACCOUNT. Feature importance is tracked, and 100 trees are used in the ensemble.



## Result

Predicting Move-in Date

The algorithm was trained on **21,549** observations derived from date of birth (DOB) data in the **Birthdate** dataset to predict **EVENT\_DATE** in the **MoveIn** dataset. A Random Forest model was employed for this task. The prediction accuracy was evaluated with a margin of error of ±**30** days. When tested on a separate set of **5,385** observations, the model achieved an accuracy of **83.5**% within the specified error range.

Predicting Move-in Premise

The algorithm was trained on **21,549** observations using the **SOURCE\_ACCOUNT** data from the **Birthdate** dataset to predict **DEST\_ACCOUNT** in the **MoveIn** dataset. A Random Forest model was utilized for this task. The training data included **5** distinct **SOURCE\_ACCOUNT** values and **5** distinct **DEST\_ACCOUNT** values. When evaluated on a separate set of **5,385** observations, the model achieved a prediction accuracy of approximately **93.47%**.