In this report, I will explain the entire data processing and modeling workflow for predicting the destination premises (`DEST_PREMISES`) based on several features from a given dataset. The steps include data loading, cleaning, feature engineering, model training, prediction, and evaluation. Below is a detailed explanation of each part of the process.

1. Loading the Data

The first step in the analysis was to load the datasets into R using the `read.csv` function. Two CSV files were read in:

- `birthday` data: Contains birthdates of individuals (TAG_START).
- `movein` data: Contains event dates when individuals moved in (EVENT_DATE).

These datasets are essential as they provide the information about the individuals' movements and their respective birthdates.

2. Date Conversion

Both the `DOB` (date of birth) and `EVENT_DATE` columns in the `birthday` and `movein` datasets respectively were in character format, so we needed to convert them into Date objects for proper handling. This was done using the `mdy_hm` function from the `lubridate` package, which converts strings to Date objects. Then, we further converted the resulting `Date` object to ensure it was in the correct format.

3. Filtering the Latest Move-In Records

For each individual identified by their `TAG_START`, the goal was to keep only the record with the latest move-in date (`EVENT_DATE`). This was achieved by grouping the data by `TAG_START`, then filtering to retain only the row with the maximum `EVENT_DATE` for each individual. This ensures that we have the most recent move-in event for each person.

4. Removing Duplicate Records

In both datasets, duplicates of `TAG_START` were removed. This is done to ensure that each individual is represented by only one record in each dataset, avoiding any redundancy.

5. Merging Datasets

The next step was to merge the `birthday` and `movein` datasets by the common key `TAG_START`, which links the two datasets together. A left join was used to ensure that all

records from the `movein` dataset were preserved, while matching birthdate data from the `birthday` dataset was added where available.

6. Feature Engineering

Several new variables were created to enrich the dataset:

- **Age at Move-In**: This variable represents the age of each individual at the time of their move-in. It was computed by taking the difference between the `EVENT_DATE` and `DOB` in days and then converting the result to years (by dividing by 365.25 to account for leap years).

- **Move-In Year**: Extracted from the `EVENT_DATE`, this represents the year when the individual moved in.

- **Move-In Season**: Based on the month of the `EVENT_DATE`, a categorical variable representing the season of the move-in was created. The `case_when` function was used to categorize the months into four seasons: Winter, Spring, Summer, and Fall.

7. Data Cleaning

After creating the necessary features, I selected the relevant columns for the modeling process, which included:

- `TAG_START`, `DEST_ACCOUNT`, `DEST_PREMISES`, `EVENT_DATE`, `SOURCE_ACCOUNT`, `DOB`, `Age_At_MoveIn`, `MoveIn_Year`, and `MoveIn_Season`. Additionally, rows with missing values in the key columns (`DEST_PREMISES`, `DEST_ACCOUNT`, and `SOURCE_ACCOUNT`) were removed to ensure the model training process is not affected by missing data.

8. Splitting the Data into Training and Testing Sets

To evaluate the model's performance, I split the data into two sets: one for training the model and the other for testing its accuracy. The training set consisted of 80% of the data, while the testing set comprised the remaining 20%. A random seed was set to ensure reproducibility of the split.

9. Converting Variables to Factors

For categorical variables like `DEST_ACCOUNT`, `SOURCE_ACCOUNT`, `MoveIn_Season`, `MoveIn_Year`, and `DEST_PREMISES`, I converted them into factors, as these are essential for

building a classification model. Factors in R represent categorical variables and are needed for the model to understand them correctly.

10. Model Training

I then trained a Random Forest model using the `randomForest` function from the `randomForest` package. The target variable (`DEST_PREMISES`) was predicted using a set of predictors: `DEST_ACCOUNT`, `SOURCE_ACCOUNT`, `Age_At_MoveIn`, `MoveIn_Season`, and `MoveIn_Year`. The model was trained using 100 trees, which is a typical number of trees in a Random Forest model for classification tasks.

11. Prediction and Evaluation

Once the model was trained, I used it to predict `DEST_PREMISES` for the test data. The predictions were compared to the actual values from the test set. To assess the accuracy of the model, I calculated two error metrics:

- **Mean Absolute Error (MAE)**: Measures the average magnitude of the errors in predictions, without considering their direction. The MAE was calculated by taking the absolute difference between the predicted and actual values and averaging over all the test cases.

- **Root Mean Squared Error (RMSE)**: Measures the square root of the average of the squared differences between predicted and actual values. RMSE is more sensitive to large errors than MAE.

While MAE and RMSE provide a numeric measure of how well the model performs, for categorical outcomes like `DEST_PREMISES`, metrics like accuracy or a confusion matrix would be more informative.

12. Saving Results

Finally, the predictions and actual outcomes for each `TAG_START` were stored in a data frame, which was then written to a CSV file for further analysis and reporting.

Conclusion

In summary, the goal of this analysis was to predict the destination premises (`DEST_PREMISES`) based on various factors such as the individual's move-in date, birthdate, account details, and seasonal factors. The Random Forest model performed reasonably well, with the calculated MAE and RMSE providing insights into the model's predictive power. The final results were saved to a CSV file for further review and potential use in decision-making processes.