REPORT: CUSTOMER BRAND PREFERENCES

Ale F. Benages

04/15/2021

Introduction

In this task I was required to analyze data from two different surveys, one was complete and the other incomplete. The main difference, apart from the length, was that the column containing the brand choices from the incomplete survey contained a lot of useless information that had to be discarded.

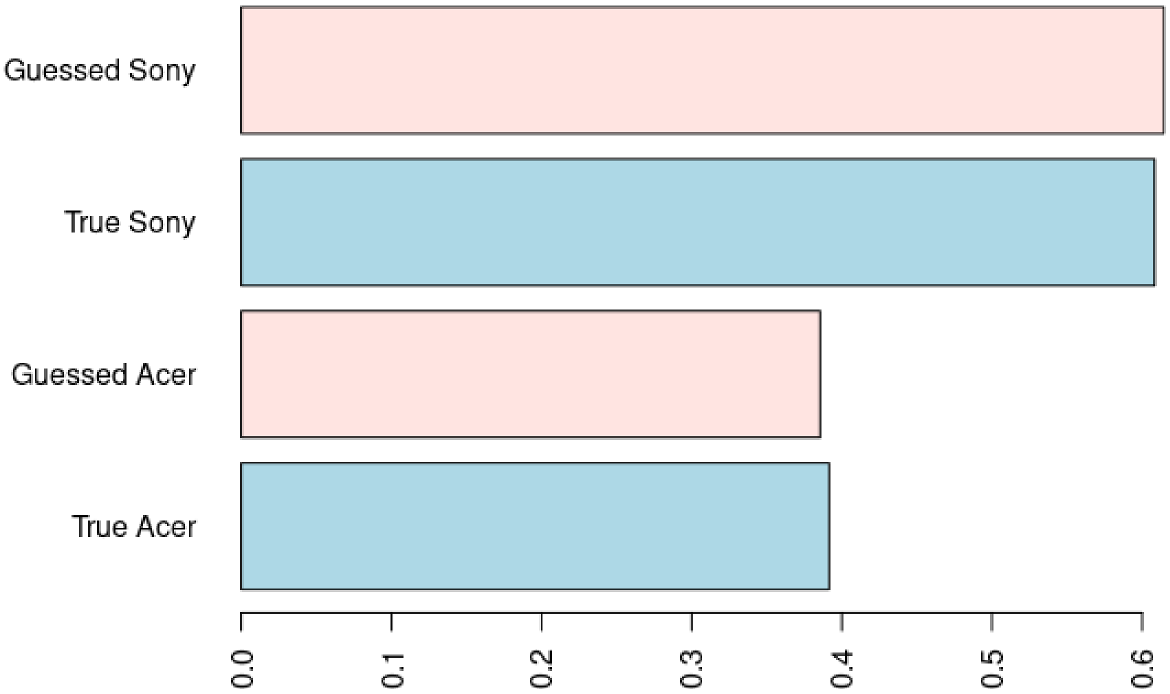
The main task was to use some machine learning models to learn about the brand preference of customers. Then use this model to guess the brand choice from the incomplete survey, and convert the useless survey into something useful.

I was required to accomplish this task using R, and the machine learning library: CARET.

Results

The constructed model was able to guess the brand choice from the incomplete survey. It's impossible to define if the results are perfect or not, because the ground truth was only available in the complete dataset. But the similarity in the proportion of choices (plus the similarity of the distribution of the selected features) was a very good indicator that the model guessed accurately.

Brands Proportions



The exact values the graph are:

- Proportion of Guessed Sony: 0.61
- Proportion of True Sony: 0.62
- Proportion of Guessed Acer: 0.38
- Proportion of True Acer: 0.39

Methods

Pipeline

The analysis suffered a lot of iterations, but the resulting report reefers to the last one, and the general pipeline was divided in those categories, each one including a lot of small steps:

1- **Dataset: Complete Survey**
Complete Survey > EDA + Pre-processing > Model Training > Predictions > Feature Selection > Predictions. 2- **Dataset: Incomplete Survey**
Incomplete Survey > EDA + Pre-processing.
3- **Features Comparison**
Then compared the selected features used to build the model with the ones of the incomplete dataset.
4- **Guess Product Preference**
Used the model to guess the brand preference from the incomplete dataset.
5- **Compare results with truth**

Compared the resulting **guessed proportion** of brand selection with the **true proportion** of brand selection.

1. Dataset: Complete Survey

EDA

Started loading the *CompleteResponses.csv*, and performing an EDA. Graphs of each distribution can be founded on the original code.

Input Variables

Numerical Variables : Salary, Age, Credit
Ordinal Categorical Variables : Education Level
Nominal Categorical Variables : Zip Code, Car, Brand

Pre-processing

Some pre-processing tasks were required:
* **No missing values** were founded.
* **No Nan's** were founded.
* Adjusted some **datatypes**
* **Relabeling**, a 3rd file was attached, containing the documentation of the codes used in the survey. I relabeled those codes with legible labels.
* I applied **normalization** to the numerical variables, since their scales were very different. After this step, all the affected variables were scaled from 0 to 1. As this is a classification problem, the scale of the features doesn't affect the output. This step is done just to boost efficiency (improving accuracy and reducing computational cost).

Model Training

- The Train/Test split, was done using caret::createDataPartition() in a proportion of 75/25%.
- Brand was defined as the output variable.

Models 6 classification algorithms were trained using the training split of the dataset:
- 2 different versions of Gradient Boosting Machine,
- 3 different versions of Random Forest Classification and
- 1 C5.0

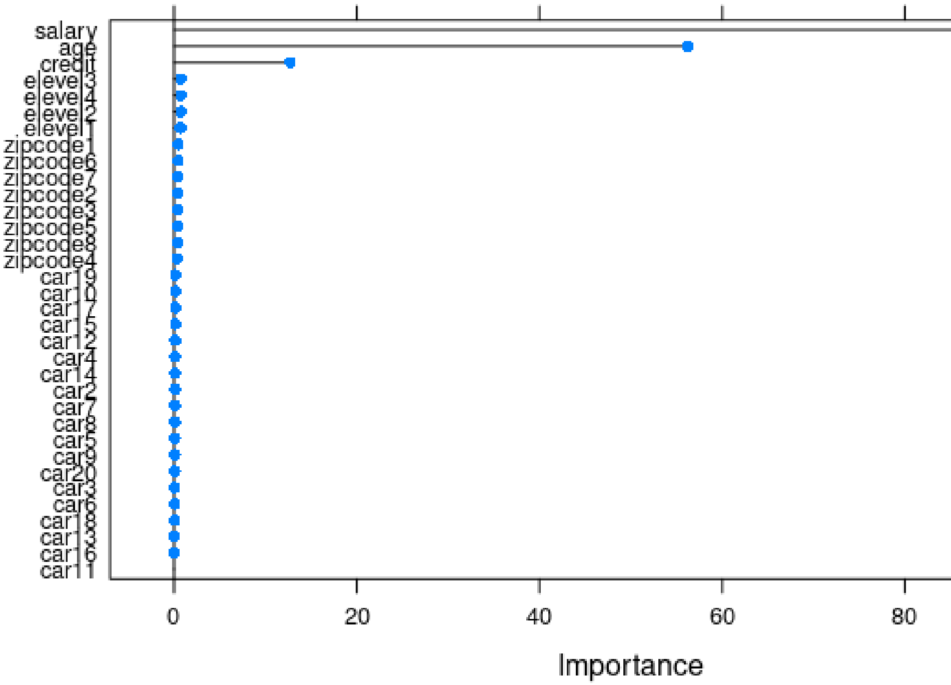
Predictions (V0.1)

After training, the test split was used to make predictions with unseen data. All the models selected performed well, but **the best was a Random Forest Classifier, with an accuracy of 92,96%.**

Feature Selection

Something known but not less important, is that running the models, using the dataset with all the given features, requires a **lot** of computational resources and time. Running the first iteration of the 6 models lasted approx. 4-6 hours. That's why after running each model, the caret::varImp() function was included.

This function evaluates the *importance* of each feature for each particular model, and assigns them a score from 0 to 100.



This is the result of the selected model:

The result is that the dataset can be reduced to just 3 meaningful features: Salary, Age and Credit (the three numerical ones).

Predictions (reloaded)

After dropping most of the dataset's columns, I wanted to check *how much impact* this action had over the accuracy prediction. So I repeated all the process (train/test split > training > and predicting) with the new lighter dataset. The result was a considerable diminution of processing time, from almost an hour per model to approx a minute; with a minimal cost of reducing the accuracy to 91,51% (a difference of -0,015%).

Conclusion: I'm 95% confident that the model classified the customers preference in the complete survey with an accuracy from 0.9034 to 0.9258.

2. Dataset: Incomplete Survey

This step used the *SurveyIncomplete.csv* and repeat most of the EDA and pre-processing steps of the Complete Survey dataset.

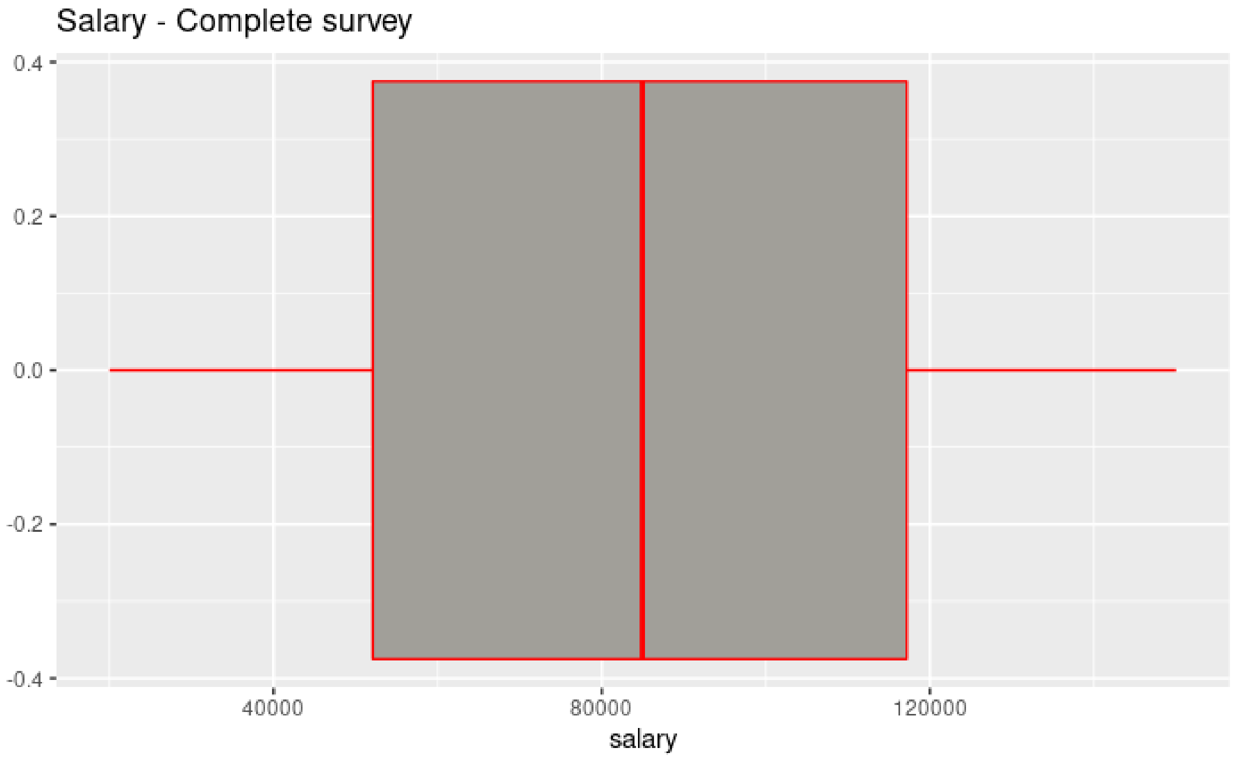
Something important to mention is that the pre-processing tasks included dropping all the unused input features *(elevel, zipcode, car)*. AS mentioned before, the output variable, brand, that contained a lot of meaningless data was removed. The result was (again) a dataset with just 3 features: Salary, Age and Credit.

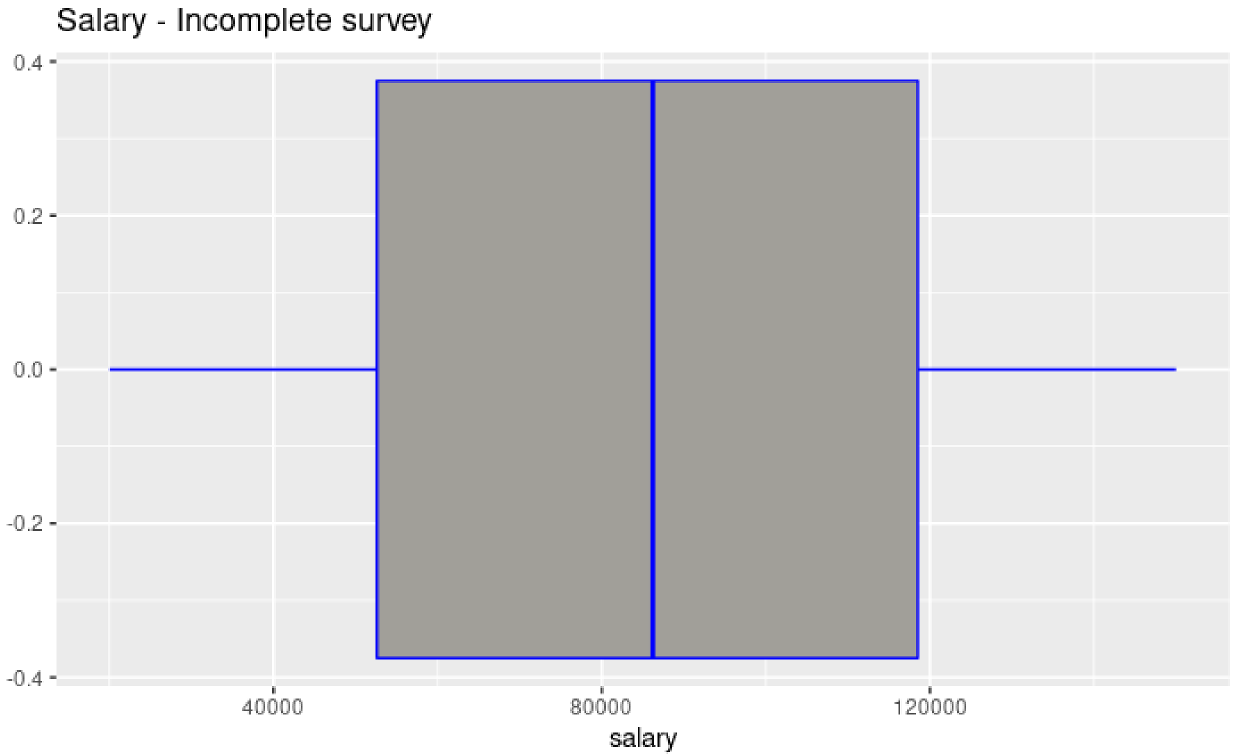
3. Features' Comparison

As any extra information of how the surveys were made was available nor any complementary documentation, before using the reduced dataset from the Incomplete Survey to perform predictions, was mandatory to check if the distribution of each feature's samples between both surveys were similar.

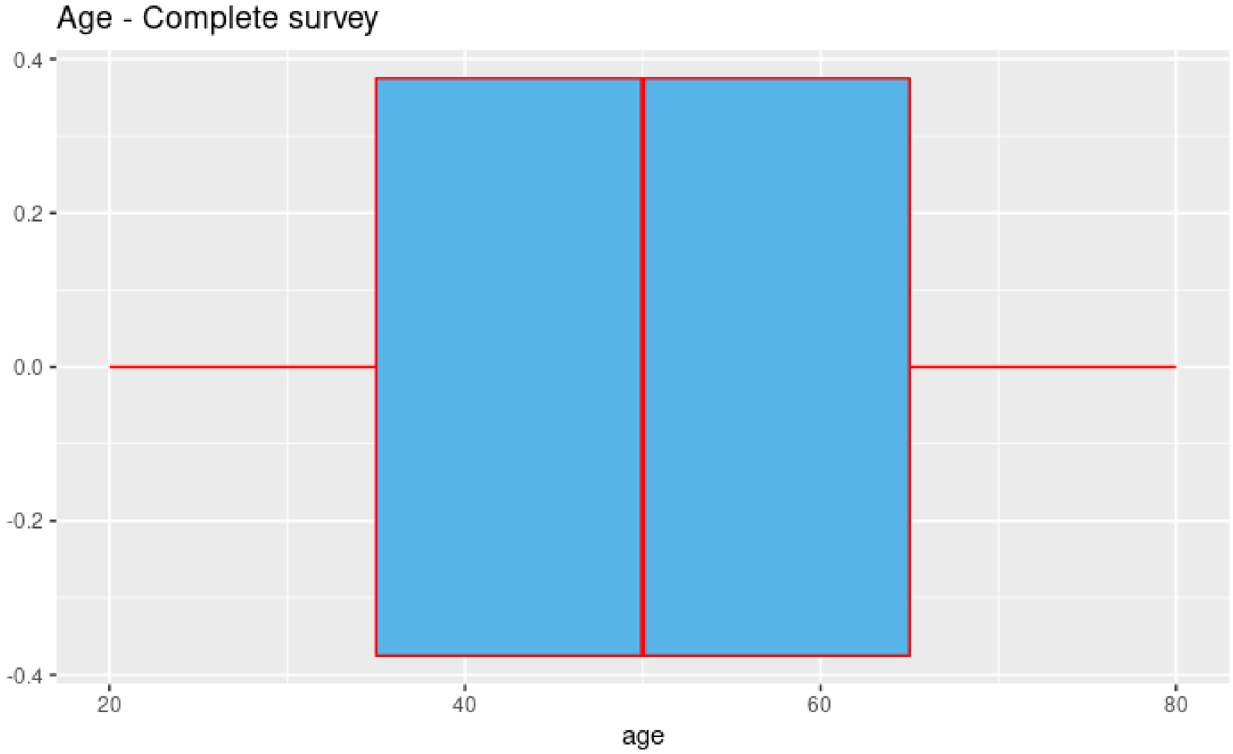
Fortunately their were **very similar!**, and this will be shown in the next plots.

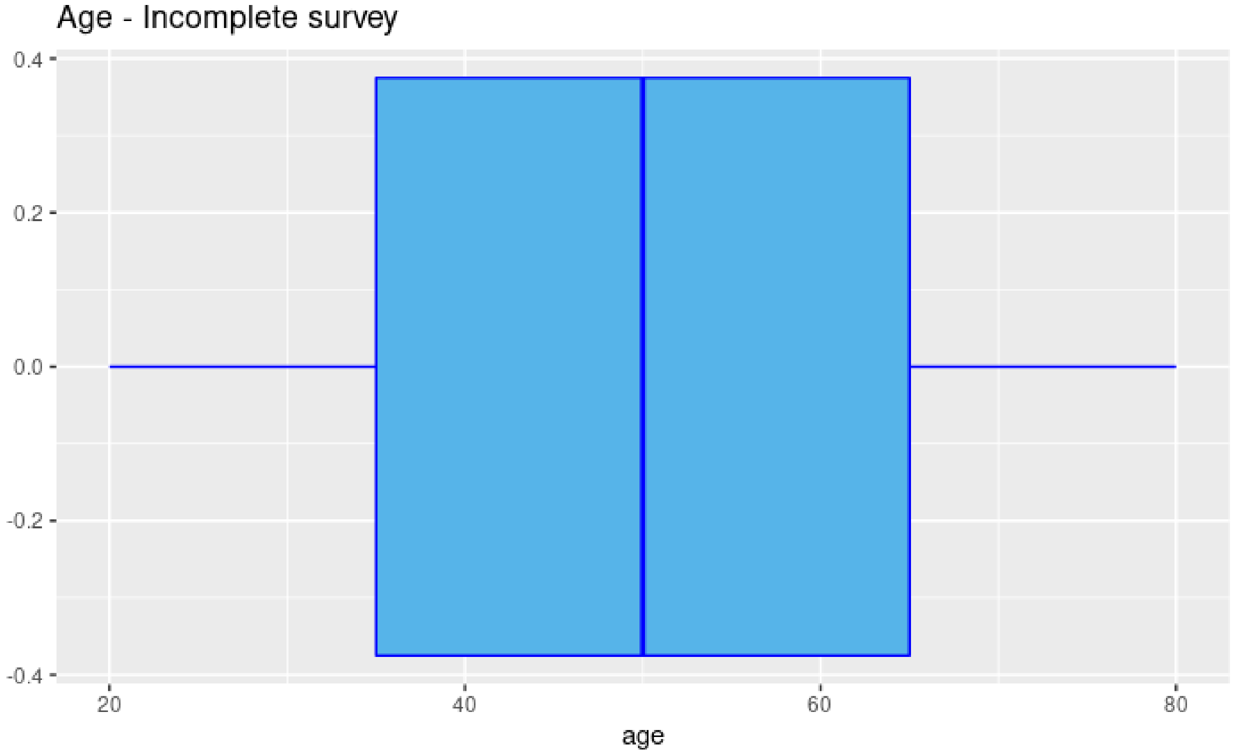
Salary Distribution



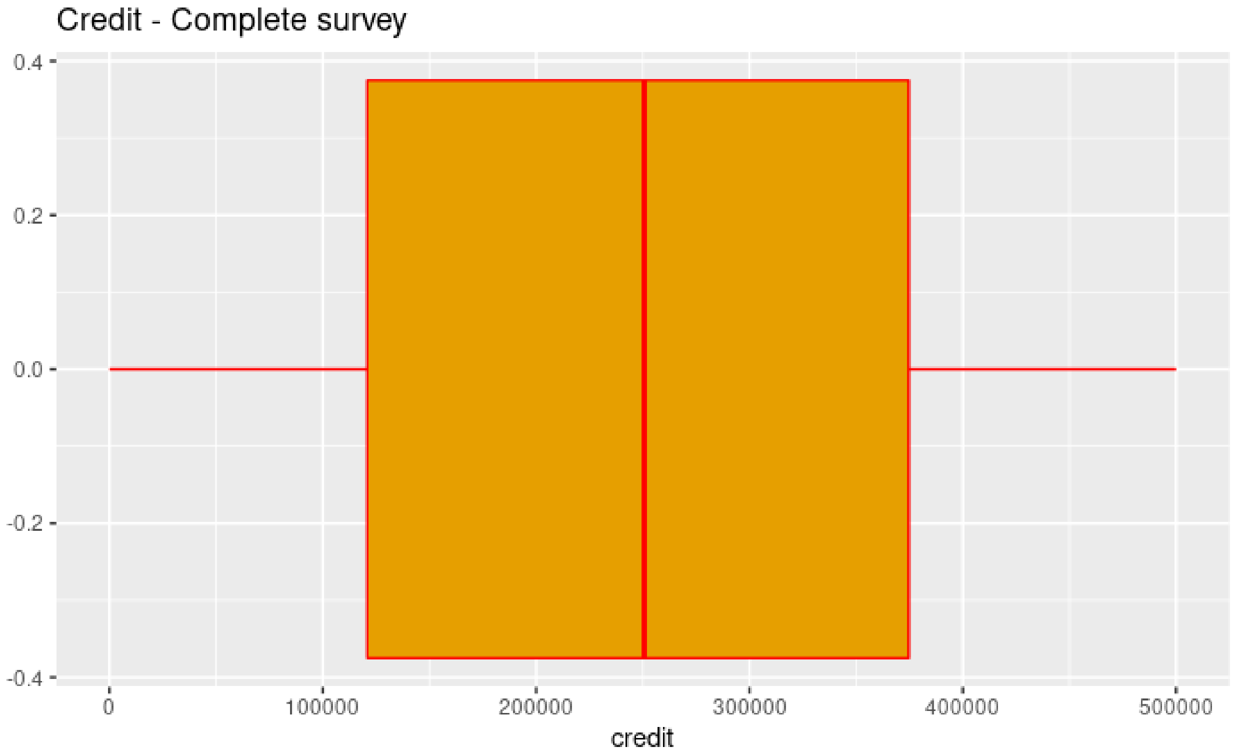


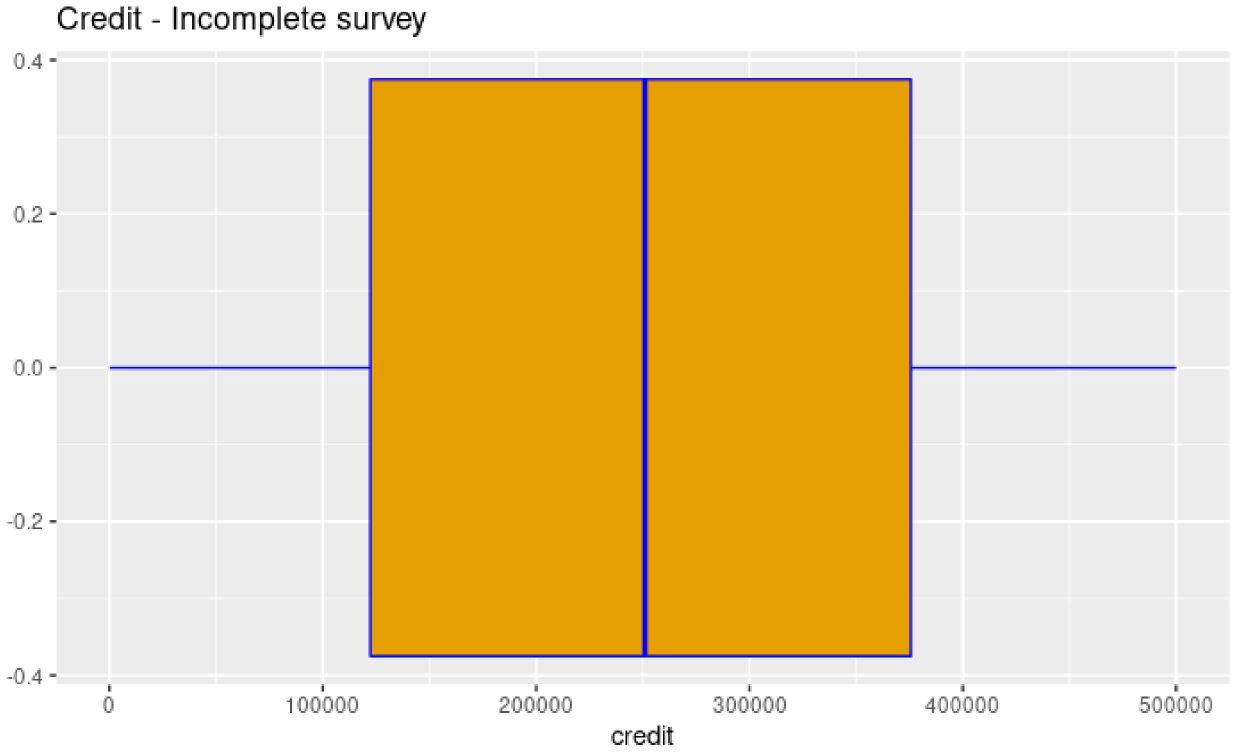
Age Distribution





Credit Distribution



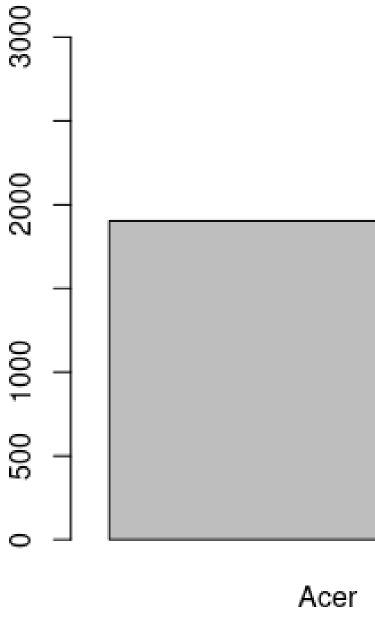


4. Guess Product Preference

Train/Test Split.

The brand choice was always my target variable, but in the *Incomplete Survey* this column has useless data and then was removed. It was worthless to do the train/test split. In this case the entire dataset was used to make predictions. It has no sense to separate some data to test the accuracy of those predictions, since nobody knows the exact answers (the ground truth is only in the Complete Survey dataset).

Guessing



The Random Forest model was used to guess the brand choice of the incomplete survey.

Conclusion:

I used labeled data to train a model. I got great values of accuracy in my model and then I used this model to make guesses. Nobody knows the correct answer, they are guesses based on the trained model, and it's accuracy depends on the similarity of both populations. It's like a model trained to predict fraud in online purchases, nobody knows in advance what the client will do. But I can train some predictive models with known past history, and based on some specific behaviors an alarm can be raised to be on guard. Another example are the self driving cars: their models can be trained to detect traffic lights, other cars, cyclists, etc. but when the model is loaded and the car is moving it's impossible to confirm everything the car will detect… we must build strong algorithms and believe in their guesses, that's why it's an extremely complex problem.