Case Study Auto Robot Company

Emiliano Isaza Villamizar

1. Introduction

The Company (Auto Robot) wants to discover unhidden patterns on its fleet of devices transmitting daily aggregated telemetry attributes. This document seeks to solve Auto Robot's inquiries. The document is organized in the following manner: first, we will state the objective then, we will pursue a Data exploration in order to comprehend the data we have available and try to further manipulate it for the sake of the model's performance. We will then create a statistical model that addresses Auto Robot's business problem and further explore future analysis and limitations of the created model. All the work with this data is available in the git hub repository: https://github.com/orangebacked/ibm-auto-robot.git.

2. Objective

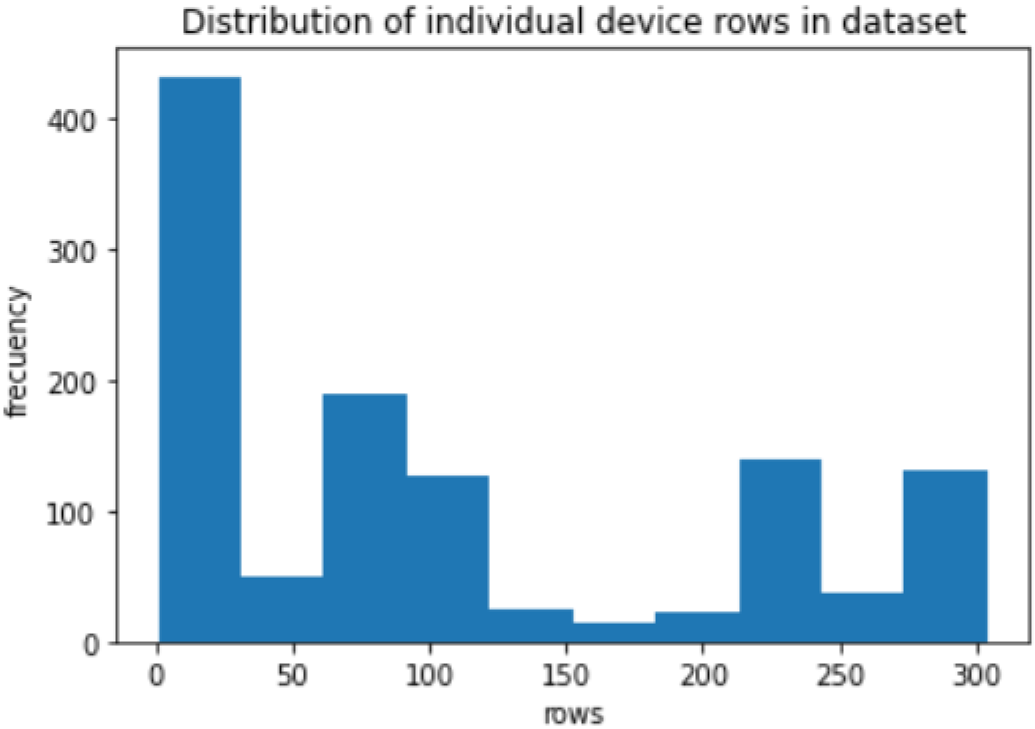
Use predictive techniques to reduce maintenance costs of devices in particular we have to build a predictive model using machine learning to predict the probability of a device failure.

3. Data exploration

In this section, we will perform an EDA on the set of data in order to gain some insight into the dataset and, hopefully, get some modeling ideas. We will try to solve three questions: how are the devices distributed in the dataset? How are the failures distributed (is the imbalance biased)? Is the time series stationary or does it exhibit seasonality?

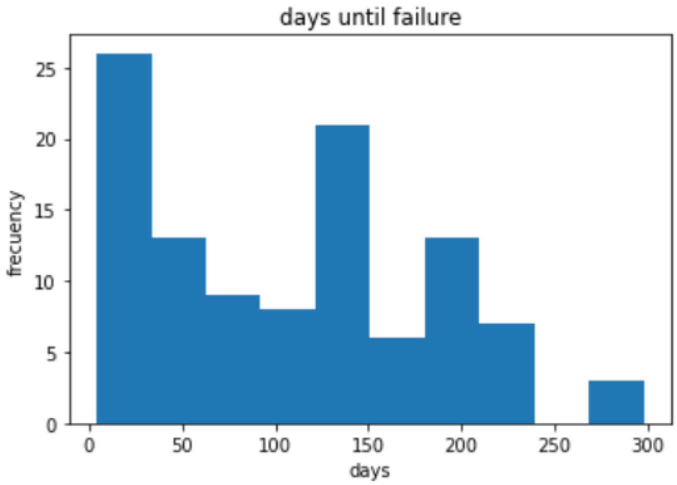
a. how are the devices distributed in the dataset?

The data shows that devices within the dataset follow a distribution, that entails that the devices that are more common in the dataset will be skewed towards these devices if this number is considerable; however, the maximum value a device is mentioned in the data is 300 which represents roughly .24 percent of the data. Therefore, we can assume the influence of these "common" devices is negligible.

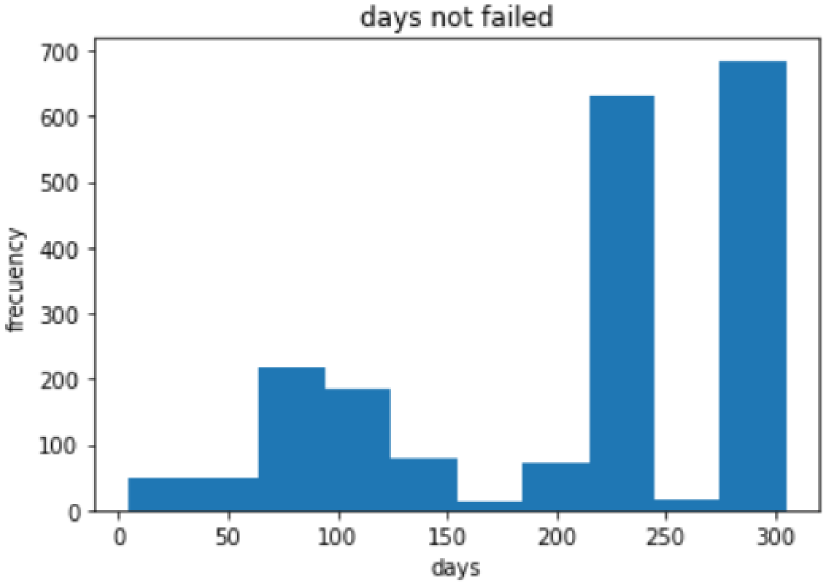


b. How are the failures distributed (is the imbalance biased)?

There are 10 times more devices that haven't failed than those that have failed, let us check the time the devices have been in the data to check our hypothesis that they are new devices, since we do not know when the device started to be used, and there is no continuous data of the IoT device, we will start checking when it was registered in the data base. The next histogram shows the distribution of days from the first error message until there was a failure for the devices that failed.



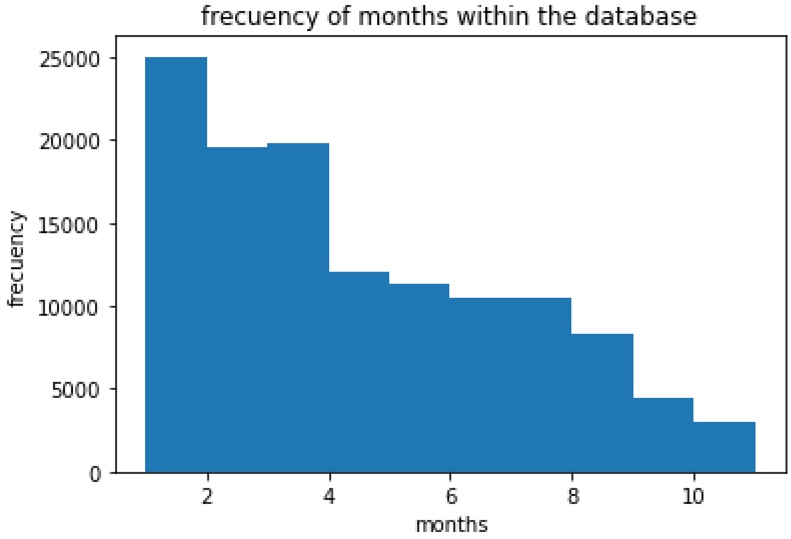
Moreover, if we check the distribution of days that have been registered of devices that have not failed we find a very counterintuitive result.



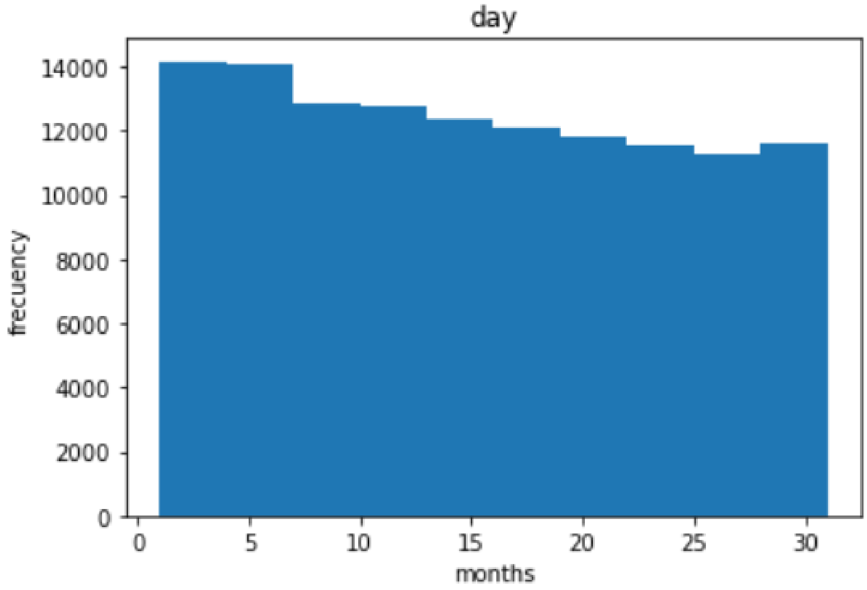
There are devices, many, that have records but do not fail for a very long time. Well, there appears to be no intuitive correlation in the time since the first error was transmitted and the probability of failure, so that hypothesis was wrong. Let us address the next question

c. Is the time series stationary or does it exhibit seasonality?

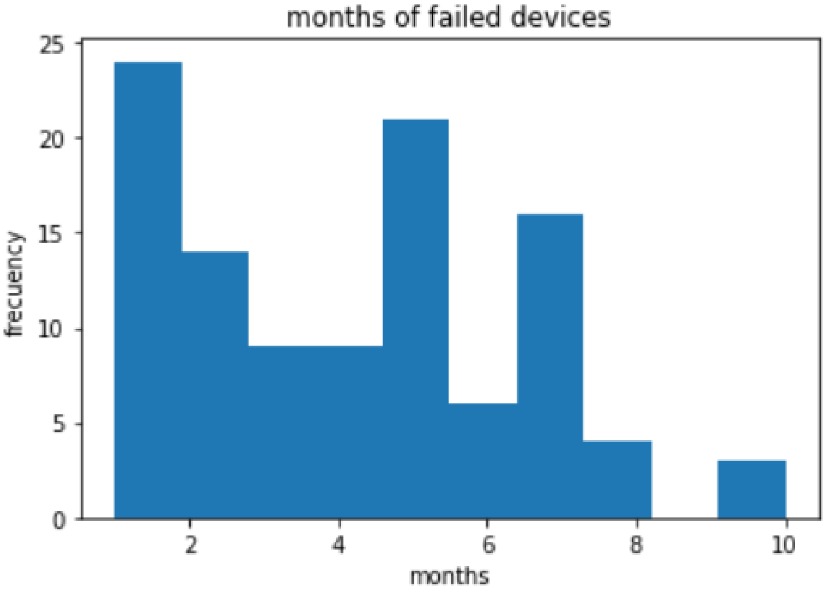
To answer this question we will revise the distribution of months and days within the data

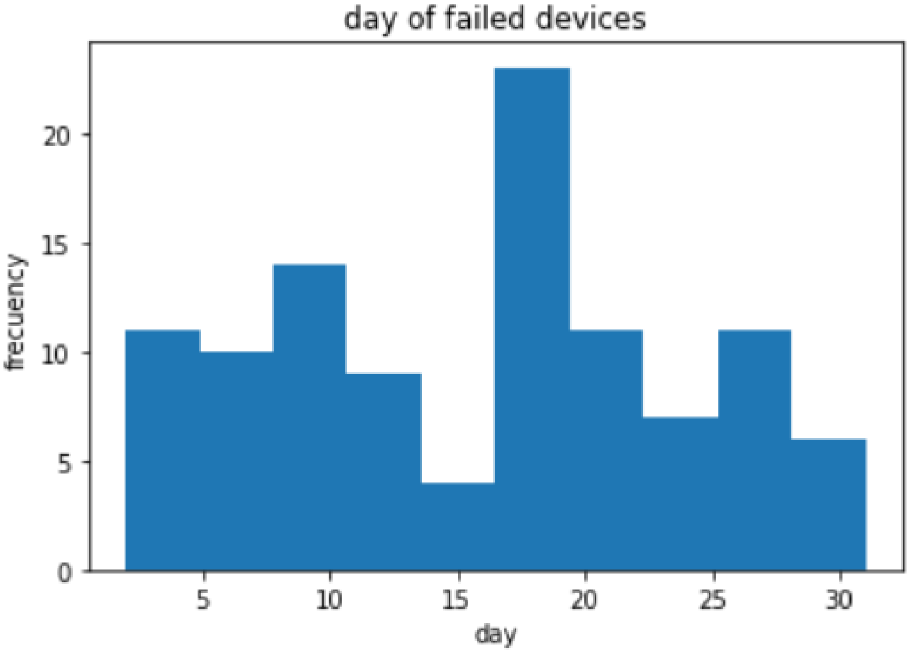


The former histogram shows that the data is skewed towards the early months of the year. Therefore, there are more error logs at the beginning of the year. In contrast, the days of the month follow a uniform distribution.

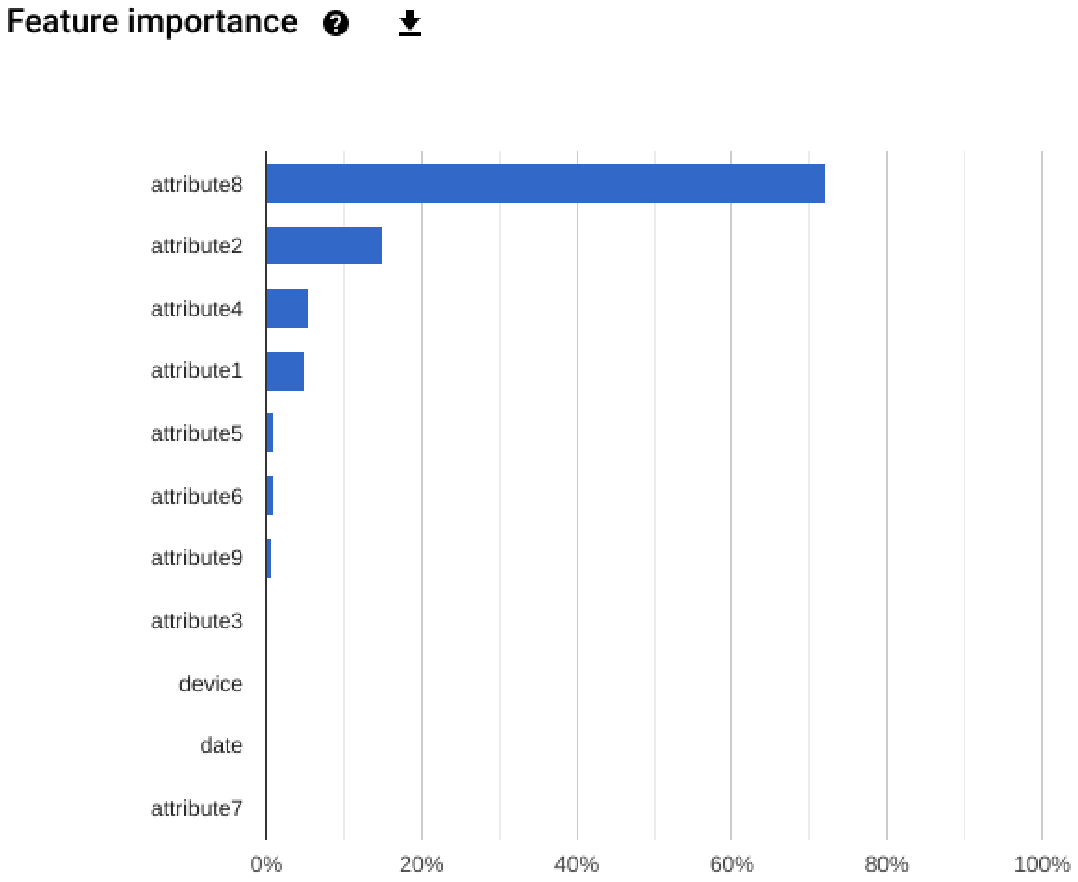


However, these distributions are not independent of the failed devices, in the following histogram we evidence that there a months that is more likely than other for devices to fail as well as more dangerous days for the devices:
https://github.com/orangebacked/ibm-auto-robot.git

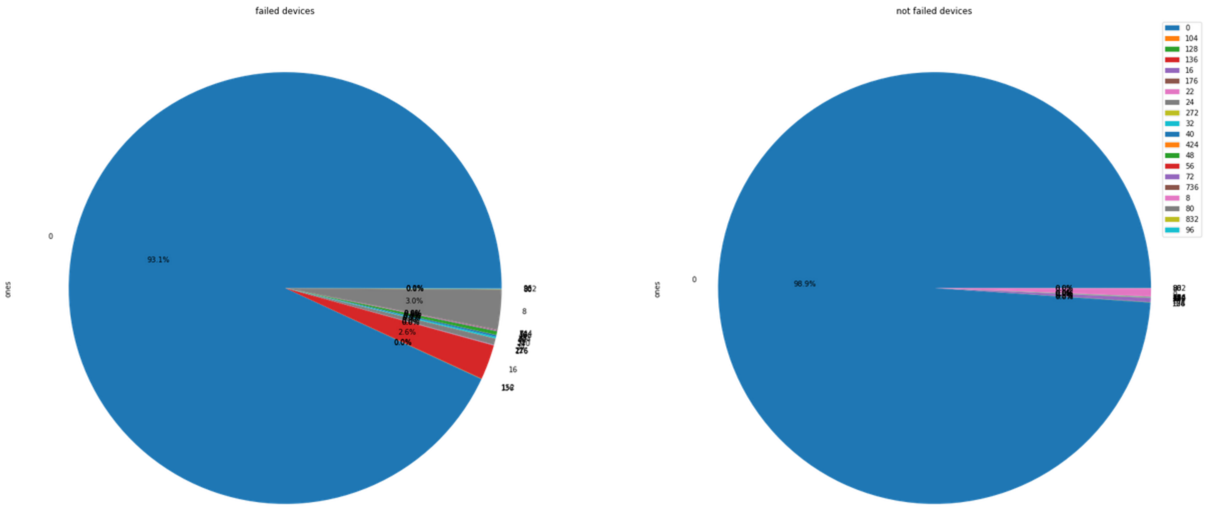




Doing some correlation tests, the attributes are highly correlated with the probability of failure:



Exploring the attribute8 we can see stark differences between the set of devices that failed and those that did not:



4. Feature engineering and data train and test creation

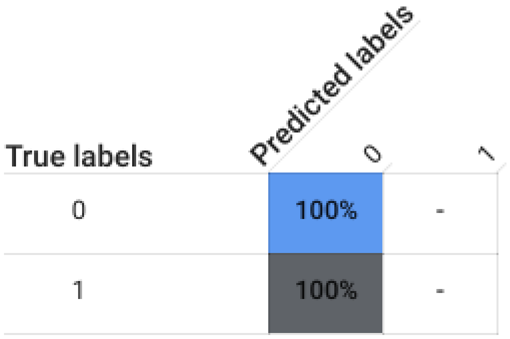
The full code is available in the git hub repository inside the file feature_eng_and_creating_dataset.py . But basically, we considered attributes 1 and 6 as continuous due to their variance and number of unique values and the rest of the attributes as categorical variables. We converted the rest of the variables into dummy variables; the resulting matrix had 15166 columns as a result of this process, we had to do a dimensionality reduction to get decent results so we applied a PCA with 95% of explained variance in order to speed up the algorithm and get some interpretable results. Before we performed PCA we used the devices that had presented failure and we chose the same number of n observation corresponding to random observations that did not present any failure, in this way I can help solve the imbalance and prevent overfitting from a simple algorithm such as the Logit we used later

5. Model exploration

a. Auto ML Google

I ran the dataset through Google auto ML and got terrible results, I believe I have to make a little bit more feature engineering and add more computational time in order to get good results. The following matrix shows the confusion matrix for the auto ml model. The model is incapable of classifying a failed device.

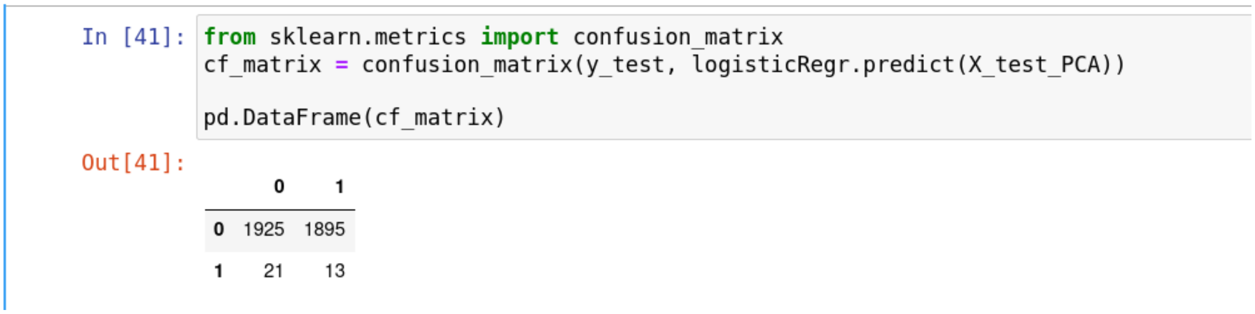
7.



	Predicted labels	
True labels	0	1
0	100%	-
1	100%	-

b. Logistic model

I ran a logistic model using the matrix I got from the PCA and got somewhat better results:



```
In [41]: from sklearn.metrics import confusion_matrix
         cf_matrix = confusion_matrix(y_test, logisticRegr.predict(X_test_PCA))

         pd.DataFrame(cf_matrix)
```

```
Out[41]:
              0     1
         0 1925  1895
         1   21    13
```

The model was able to predict 13 true positives out of 34 giving me a false positive rate of 61% and a false negative rate of 49%

6. Further recommended analysis

7.

The model fails to produce results that could be attainable. Further exploration and feature engineering is required in order to successfully model the probability of failure of the devices made by the Auto Robor company. Moreover, there should be data from every device not only if it sends an error there is a biased in the data due to the inability to measure how many days the device has been in service. In particular, the days between the first error and last error histogram is very counterintuitive. We suggest further improvement to be made in the recollection of data to have a balanced panel data, this will no doubt have great effects on the prediction results; moreover, we need to explore more strategies and algorithms to achieve better prediction results.