EDA & Baseline Model

This part of the work is all located in one repository, and contain:

1. This pdf, organized by sub-subjects
2. Doc version of this pdf
3. Main Jupyter notebook, with most of the code and the outputs, named “EDA and Baseline Model.ipynb”
4. Two automatic reports:
   a. Pandas Profiling ("pandas tool part report.html") – created in the Main Jupyter notebook.
   b. Mechkar (“mechkar tool part report.html”) – created in little separated Jupyter notebook written in R.
5. The R notebook for creating the “Mechkar” report, named “mechkar.ipynb”.
6. “ff.csv” – csv version of the labeled dataset (flat file) – relevant primarily for loading the data in R.

EDA

1) Target variable insights

Analyzing the target variable (“Value”) is in some ways, act that happens almost in every operation in the EDA. However, those are the main contexts of it:

a. Under “General look on the target variable” – cells 13-14 in “EDA and Baseline Model.ipynb”. in those cells we learned the distribution of “Value” – 66% for “0’ label (no touch) and 34% for “1” label (touch).
b. Under “Dependent Variable Distribution” in “mechkar tool part report.html” we can see the distribution of each variable for each label of the “Value”. I will analyze it in the feature analyses section.
c. Analyze of connections between specific features and “Value” – in cells 15-32, 39-50. Also, will be discussed later.
d. Each feature’s correlation with the target variable, in cell 36. As mentioned above, will be discussed later.

2) Data quality

Already in cells 9-10, I checked for number of missing values in each column, when the number of zero values can be a clue for practically missing value. I learned that there are almost no missing values, besides of few columns that have only missing values.

In “pandas tool part report.html", in the “alerts” section, I met again those columns, but not just them. I also realized from that section. That there are some other columns with constant value, like INSERT_REPLACEMENT_REPORTED that has constant value "False".

All of those columns that include only one unique value or missing any values at all, will not be relevant to analyses or training a model.

Another issue relevant to data quality is outliers. In “mechkar tool part report.html”, Under “Outliers” column, we can see visualization of the outliers’ values next to the “regular” values for each variable of the data, and also the number of outliers for each.

In cells 56-59, I created for each variable two kinds of visualizations: one that compares the variable's distribution with the same variable's distribution in case we deleted its outliers. And the other one, that compares the target variable's distribution, when each variable has outliers with the case it hasn't.
From all of those visualizations, we get pretty complexed picture. Some features have negligible number of outliers, but others, like some of the "BAND_ARR_" features, have about 1/7 of there values as outliers. Regarding to the effect of outliers on distribution, we can see variables like "Bit Flag" that the effect is obvious, but for most of the variables it's hard to notice, including variables that have a lot of outliers, like "BAND_ARR_9". On the other hand, the same "BAND_ARR_9" is example for case in which the existence or deletion of outliers, is very effective on the target variable.
This complexed picture raises the question: are effects of the outliers on the variable distribution and on the target variable, are "good" or "bad"? In other words – In what case should we delete outliers?
There are some approaches for this question. After a thought process and out of familiarity with the data, my decision in not delete any outliers. We don't have specific reason to believe the outliers are result of error in the process of producing the data, so in case that deletion changes distribution and effects target variable – those effects might be relevant and even important. In addition, we have specific orientation in this mission to notice edge cases. The analyses that have been done might help us understand the data in new ways, but will not obligate us to actual steps.

3) Feature analyses

I decided to unite sections 2 and 4 from the brief document, because some of the answers are related to each other.
A comprehensive list of v variables' types can be seen in cell 28. In "mechkar tool part report.html", in the Descriptive Statistics column, the type is more specific – not just int/float, but also continuous/categorical if needed. Summary of types can be seen in "pandas tool part report.html" in the overview section:

Categorical	6
Numeric	58
Unsupported (Nulls)	1
DateTime	2
Boolean	2

Distribution – visualization of each variable's distribution can be seen In "mechkar tool part report.html", in the distribution section. Each variable distributes differently, So I will only give few examples:

- INTERVAL_IN_FILE – in the beginning there are a lot of values, and with time there are less and less. We can learn from it about the amount of samples per machine's part – in all cases there are at least few samples, but for some machine's parts there are more samples and for other more.

- Most of BAND_ARR variables have a lot of values close to zero, almost no values later, and small hill of values later. We can learn that in most cases frequency is low, but there are cases that it's much higher, more cases than it's close to average.
- VAA has almost distribution. It can be seen in the graph, and also from the fact it's median value is almost the than its mean.

The big question of the feature analyses is likelihood of each feature to be significant. To answer this, I made few actions:

- Creating visualizations and tables, which relevant to the relationships of the variables with the target variable – mostly in cells 15-32, 39-50.
- Creating correlation metrics and tables, in cells 33-38.
- Running feature importance function in cells 63-70.
- Running feature selection functions and summaries their results – cells 77-92.

From the visualizations and correlations, I got the following insights:

- A lot of variables have strong connection to the target variable.35 variables have at least 0.64 correlation with it. Also the visualizations shows differences in the target variable distribution for a lot of variables
- There are also strong correlations between a lot of the variables one to another. From that reason, we can't conclude that strong connection to the target variable means importance for prediction.
- Specifically, cell 37 shows us the correlations between the BAND_ARR variables. It's very noticeable that some of them might be very connected to the target variable, but can't improve the predictions, since the connection is result of other variables' connection.
- Therefor, processes of feature importance and feature selection are very important.
- There are some categorical variables, that each label of them reflects in very different distribution of the target variable. That can be very useful while training tree-based models.

One variable that I was specifically asked to analyze and create visualizations for it, is SESSION_INFO_ID. That was done in cells 15-32. I would like to share few bullet points about this process:

- SESSION_INFO_ID is categorical variable with 281 unique values.
- From that reason, it's very challenging to visualize it.
- One way to handle this challenge, is to use tables and other methods instead of visualizations. In cells 15-17 that's what I did. I learned that almost half of the unique values, are sessions with no touch (value = 1) at all. That can be very helpful in predictions, primarily if we use tree-based model.
- Another way to handle the challenge is to choose the labels that have the largest number of values, unite all others as "other", and visualize from that point of view. That was done in cells 18-23. However, still most of the values were in the "other" label, so it wasn't very efficient.
- Last way to handle the challenge is to unite values into categories under the asked variable, and to visualize it. In this case, I could choose some of the categorical variables from session info table. In cells 24-32 I did it, and (as described more

generally earlier) discovered strong connection to the target variable, and also between variables one to another.

Running feature importance (cells 63-70) brought to my attention the seemingly big importance of BAND_ARR_2, and seemingly little importance of few other features, while most of the features were tagged as not important.
I continued to feature selection (cells 77-92), ran 4 methods, and summarized the results. From this part, I learned that a lot of features can be deleted before training a model, but not as much as can be concluded from the feature importance.

4) Data preprocession

Data preprocession had three main steps:

- First, in cell 60, I reloaded the data without all columns that I knew from the "data quality" section that are not relevant. I chose to reload the data from the server, and not just dropping columns, because in the EDA process I made few minimal changes that I didn't want to keep, and wanted to be sure that the data is fixed. In cells 61-62 I check that all the columns that contain only missing values, are indeed gone.
- In cells 71-76 I made some data engineering. S_I_CREATED_AT, as datetime variable, is not very helpful as is. I created from it 3 variables: month, day and hour. I thought it's likely that at least one of those will be helpful. I didn't create an year variable, because all created at 2022. I believe that later on, if new sessions will be created, the year can be important feature.
  I also thought the BAND_ARR variables have more potential. I created 3 features from them: their mean, the difference between max value and min value, and the ratio between max value and min value.
- Cells 93-98: After summarize the data selection process, I created two new dataframes. Each of those, contains less features than the original dataframe, while the features with most probability to help in the training the model, were selected to be kept. One of the new two dataframes contains less features than the other, based on the same principle. All and all, we have three dataframes, ready to be trained; as much as the dataframe has less feature, the features are likely to be more efficient. My thought is to check the baseline model for each of those dataframes, to see if data selection was helpful or not.

Baseline Model

In cell 92, I checked which of the methods I used for feature selection, selected the largest number of features. I thought it's good sign, that relevant to the selection of the model: model who uses a lot of features, tend to be more stable.
I saw SVM selected 39 features, which is much more features than the other methods selected. I thought also to train the model, from the reason I mentioned, and also because I

read it's good model for classification. However, I decided not to use SVM for the baseline model, for two main reasons:

- It's very slow to train, and I already decided to train the model for 3 dataframes
- I also read it's not working well for overlapping classes. I didn't know this term, and not sure I understood it after I read about it, but as I understood, it seems that our data might be included in this definition.

After deciding not to use SVM, I returned to cell 92, and saw the next option is lasso model. As I understand, lasso model is relevant mostly to regression problems.
Random forest model was selected, not only because I thought the previous options will not be relevant, but also because it has some advantages:

- In the feature analyze section, I already noticed that tree-based model will be efficient for some of the variables, including session info id.
- It's very quick to be trained
- It considered as "generic" default model, so for any context of explaining our choices, it raises the smallest amount of questions.

Therefore, in cells 99-117, I trained the random forest model on the three dataframes. I evaluated the results by four validation metrics: F1, Accuracy, Precision and Recall, and summarized the result into a table. In my Plan and Goal document I wrote I will want to use the F1 metric, but I thought it's too easy to run few metrics, just to be sure.
All the results are good, and there are no big gaps between the three dataframes. However, it still seems that the dataframe that had the toughest feature selection, has the best results.

<u>Next Steps</u>

The main goal of the project is to train the best possible predictive model. Our next steps should focus this goal. I think I should think and explore in four main directions:

- More feature engineering: for this stage, I created only 6 new features, related to two subjects within the data. The potential is much bigger than that, and I should explore new ideas.
- Try other combinations of feature selection: for now, I tried three combinations, while the results showed that less is more – deleting feature gives better results. Does more feature selection will upgrade the scores? Maybe other metrics of feature selection will give me new insights and new combinations?
- Most important – choose the model. From our conversations, I know the expected model should be deep learning model. However, I will want to discuss again and think : does dataframe of 114,271 rows and few tens of columns, is big enough for efficient deep learning model? Maybe we should use machine learning model for this point, and deep learning model only when we have more data?
  However, if we the decision is indeed deep learning model, I will have to engage time in understanding the different relevant models, and understand what is most suitable for our data.

- Choosing the hyperparameters for the model; using functions which check the best hyperparameters for the specific data, and by trial and error.

For conclusion, predicting the touch label is complex challenge. At this point of time, we already know that model with nice scores is possible. However, this is just the beginning; we can enrich the data and select its feature better; we can choose better model and choose its hyperparameters meticulously; and also – more data should come in the future, and it also would help us be precise.