## Main objective of the analysis that specifies whether your model will be focused on prediction or interpretation and the benefits that your analysis provides to the business or stakeholders of this data.

The main objective of the analysis here is to focus on prediction. We want to build a model that will determine if a line is faulted and what type of fault it is to better deploy technicians with the right tools to enable minimum downtime.

This project directly benefits the business by creating tripwires for response generations immediately when a fault occurs instead of waiting for an outside report.

It also creates a more informed on-site technician. With knowledge of the fault type the technician is able to plan, gather appropriate tools and the required number of people. This is different from the current method of, show up to site, evaluate, plan, return to shop and gather tools, then return to site to fix the fault.

This tool also assists the plant managers. If a fault is observed, they can take immediate action on the distribution of power to ensure zero downtime for customers.

Subtasks:

- Do some EDA on each of the features and compare the non-faulted/faulted values.
- Use Logistic regression as a baseline for classification

Possible snags:

- There may be too many features that are closely related to get accurate classification
- A less accurate model is more important than an overfit model for the use case we are trying to satisfy.

## Brief description of the data set you chose, a summary of its attributes, and an outline of what you are trying to accomplish with this analysis.

The data set I chose was Electrical Fault detection and classification from Kaggle. It measures power lines voltage and current then displays if that line is faulted and if it is what kind of fault it is.

The dataset has the following attributes:

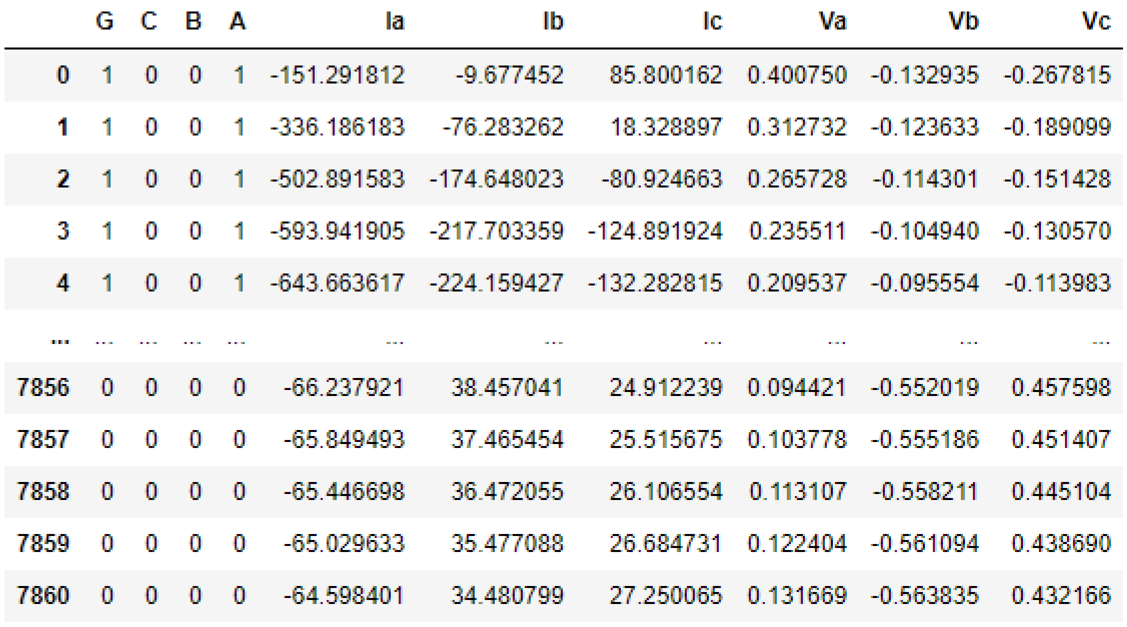
- Four different feature columns to specify the exact fault
    - Between Phase A and Ground
    - Between Phases A,B and Ground
    - Between all three phases
    - And more similar variations
    - These are indicated by 0/1 values in various columns
- The Line Current for all three phases
    - Measured in ampere
- The Line Voltage for all three phases
    - Measured in volts

I am trying to use the Line Current and Line Voltage to predict the fault category column. I believe that there will be some if not a large variation in the current/voltage between the faulted and non faulted lines.

## Brief summary of data exploration and actions taken for data cleaning and feature engineering.

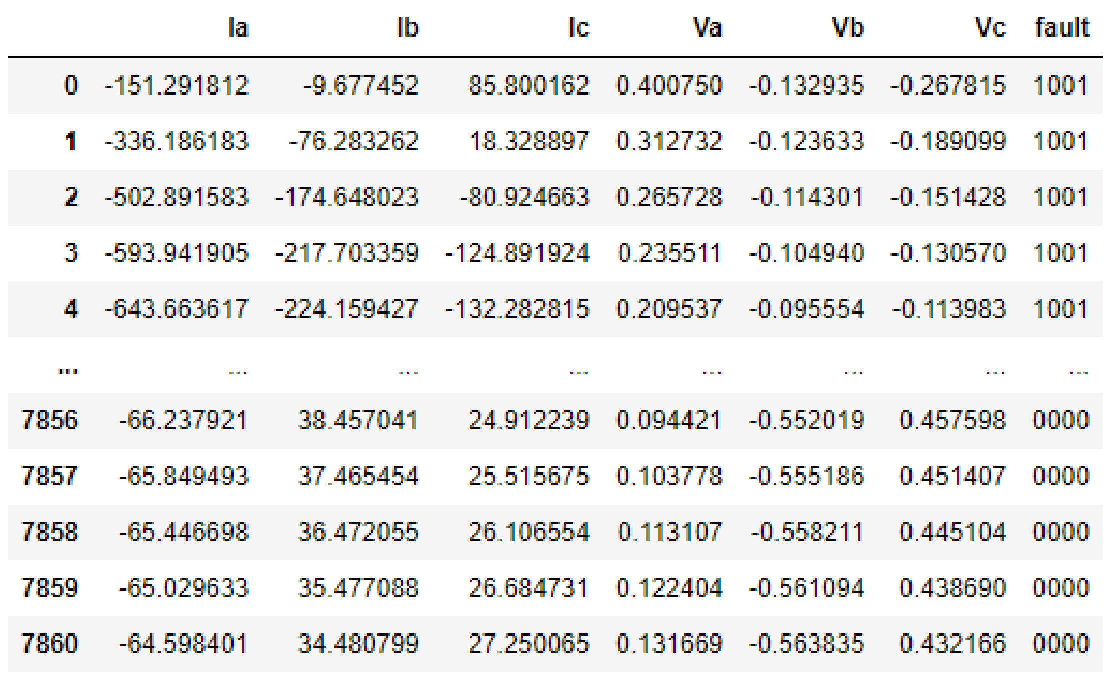
Our data was clean with no missing values thankfully. We decided to take the original fault format and condense it into something easier.

We went from:



| | G | C | B | A | Ia | Ib | Ic | Va | Vb | Vc |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 0 | 1 | -151.291812 | -9.677452 | 85.800162 | 0.400750 | -0.132935 | -0.267815 |
| **1** | 1 | 0 | 0 | 1 | -336.186183 | -76.283262 | 18.328897 | 0.312732 | -0.123633 | -0.189099 |
| **2** | 1 | 0 | 0 | 1 | -502.891583 | -174.648023 | -80.924663 | 0.265728 | -0.114301 | -0.151428 |
| **3** | 1 | 0 | 0 | 1 | -593.941905 | -217.703359 | -124.891924 | 0.235511 | -0.104940 | -0.130570 |
| **4** | 1 | 0 | 0 | 1 | -643.663617 | -224.159427 | -132.282815 | 0.209537 | -0.095554 | -0.113983 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **7856** | 0 | 0 | 0 | 0 | -66.237921 | 38.457041 | 24.912239 | 0.094421 | -0.552019 | 0.457598 |
| **7857** | 0 | 0 | 0 | 0 | -65.849493 | 37.465454 | 25.515675 | 0.103778 | -0.555186 | 0.451407 |
| **7858** | 0 | 0 | 0 | 0 | -65.446698 | 36.472055 | 26.106554 | 0.113107 | -0.558211 | 0.445104 |
| **7859** | 0 | 0 | 0 | 0 | -65.029633 | 35.477088 | 26.684731 | 0.122404 | -0.561094 | 0.438690 |
| **7860** | 0 | 0 | 0 | 0 | -64.598401 | 34.480799 | 27.250065 | 0.131669 | -0.563835 | 0.432166 |

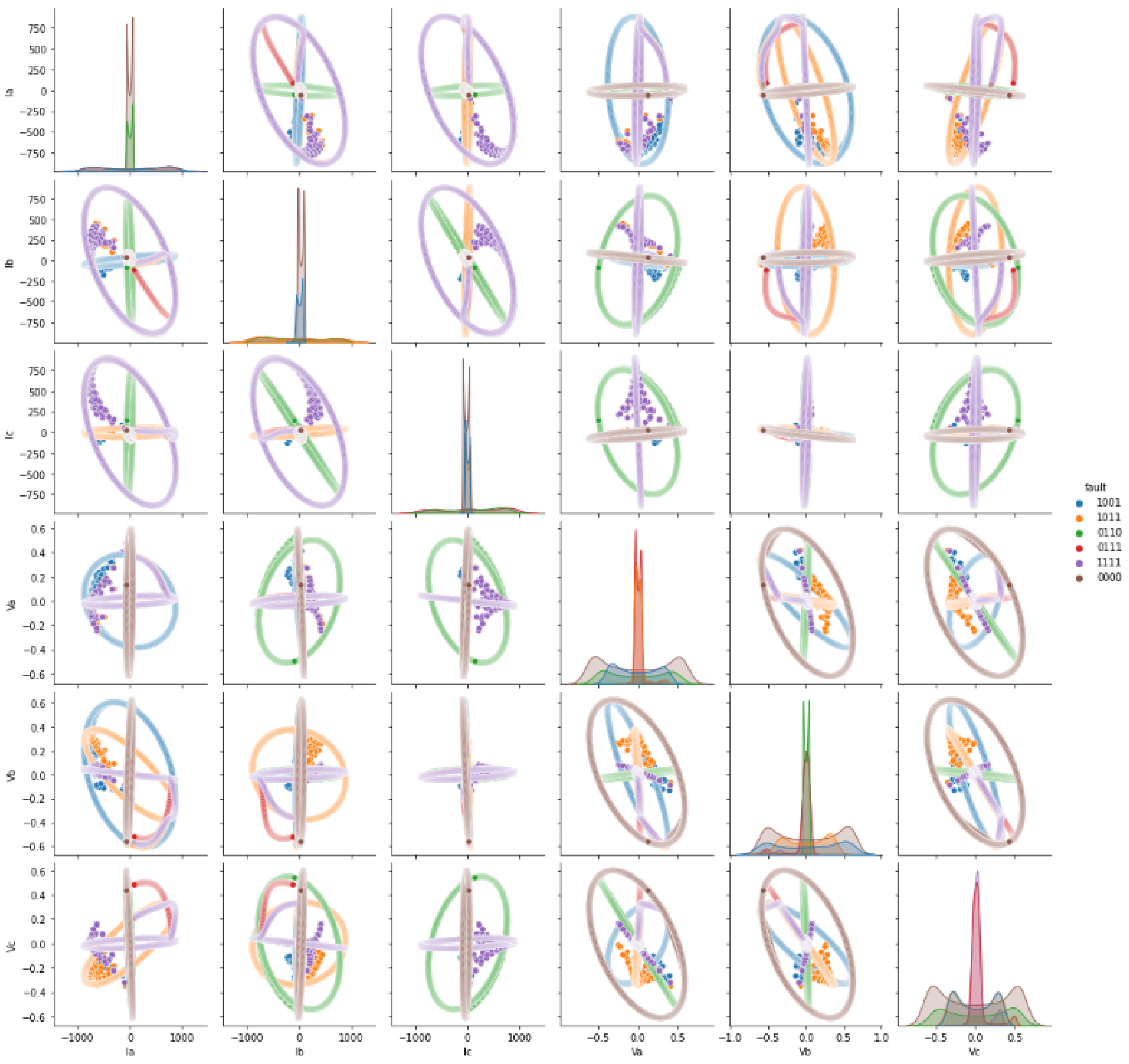
To:



| | Ia | Ib | Ic | Va | Vb | Vc | fault |
|---|---|---|---|---|---|---|---|
| **0** | -151.291812 | -9.677452 | 85.800162 | 0.400750 | -0.132935 | -0.267815 | 1001 |
| **1** | -336.186183 | -76.283262 | 18.328897 | 0.312732 | -0.123633 | -0.189099 | 1001 |
| **2** | -502.891583 | -174.648023 | -80.924663 | 0.265728 | -0.114301 | -0.151428 | 1001 |
| **3** | -593.941905 | -217.703359 | -124.891924 | 0.235511 | -0.104940 | -0.130570 | 1001 |
| **4** | -643.663617 | -224.159427 | -132.282815 | 0.209537 | -0.095554 | -0.113983 | 1001 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **7856** | -66.237921 | 38.457041 | 24.912239 | 0.094421 | -0.552019 | 0.457598 | 0000 |
| **7857** | -65.849493 | 37.465454 | 25.515675 | 0.103778 | -0.555186 | 0.451407 | 0000 |
| **7858** | -65.446698 | 36.472055 | 26.106554 | 0.113107 | -0.558211 | 0.445104 | 0000 |
| **7859** | -65.029633 | 35.477088 | 26.684731 | 0.122404 | -0.561094 | 0.438690 | 0000 |
| **7860** | -64.598401 | 34.480799 | 27.250065 | 0.131669 | -0.563835 | 0.432166 | 0000 |

## Brief summary of data exploration and actions taken for data cleaning and feature engineering. (Continued)

First observation: The distribution of faults was relatively equal around 1,100 with the exception of the “No fault condition” with 2,365 values.



Second observation: Here you can see that there are instances where a fault in a separate category can also exist.
For example: Fault 1001 is between A and Ground. But in the plot we can see under "Phase A current" you have Phase B/C faults.

Final observation: The first column in the fault code is for "ground", the second is phase C, third phase B and fourth phase A.
So the code 1001 is a fault between Ground and Phase A. Or 0110 is a fault between Phases B and C.

## Summary of training at least three different classifier models, preferably of different nature in explainability and predictability. For example, you can start with a simple logistic regression as a baseline, adding other models or ensemble models. Preferably, all your models use the same training and test splits, or the same cross-validation method.

**Logistic regression:**

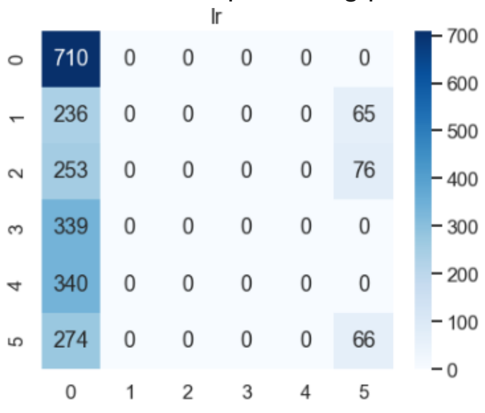
For Logistic regression we ran three different models, base LR, LR with a penalty on L1 and LR with a penalty on L2 while keeping all other values the same. We scaled the data, encoded our fault column, and did a Stratified Shuffle Split due to the imbalance across the six classes.

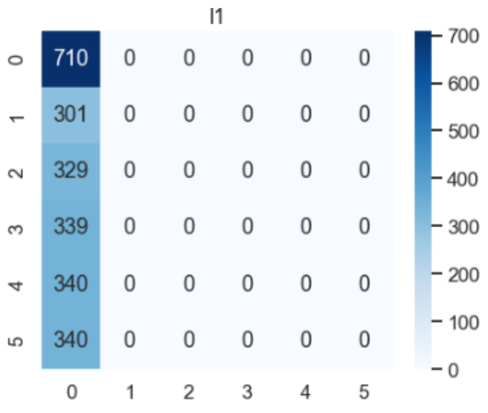
All models performed terribly. With the model with the L2 penalty performing slightly better but still not up to the standards we need.

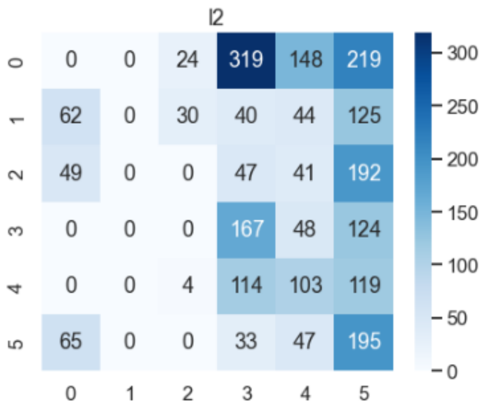
Here are the scores:

| | lr | l1 | l2 |
|---|---|---|---|
| precision | 0.145254 | 0.090586 | 0.096631 |
| recall | 0.328953 | 0.300975 | 0.197117 |
| fscore | 0.184111 | 0.139259 | 0.126610 |
| accuracy | 0.328953 | 0.300975 | 0.197117 |
| auc | 0.583693 | 0.500000 | 0.563600 |

Here is a heatmap showing predicted values versus actual values for all models:



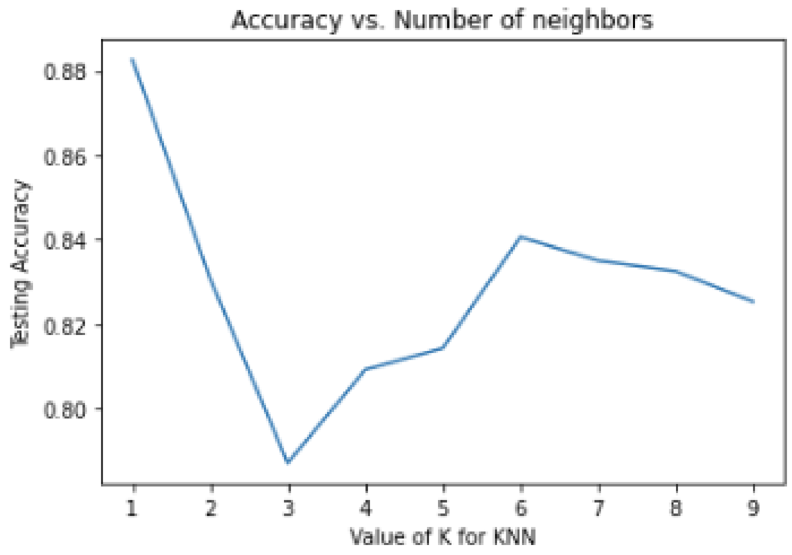




We can see that the “LR” and “L1” models struggled by classifying essentially everything as a class 0 or no fault condition. The “L2” model performed better but was still not good enough. We tried increasing regularization but just ended up classifying everything as “no fault like the other models. When we decreased regularization the model started to also falsely classify in favor of the “no fault” group. We believe that this is the limit of the base Logistic Regression model.

**KNN**:

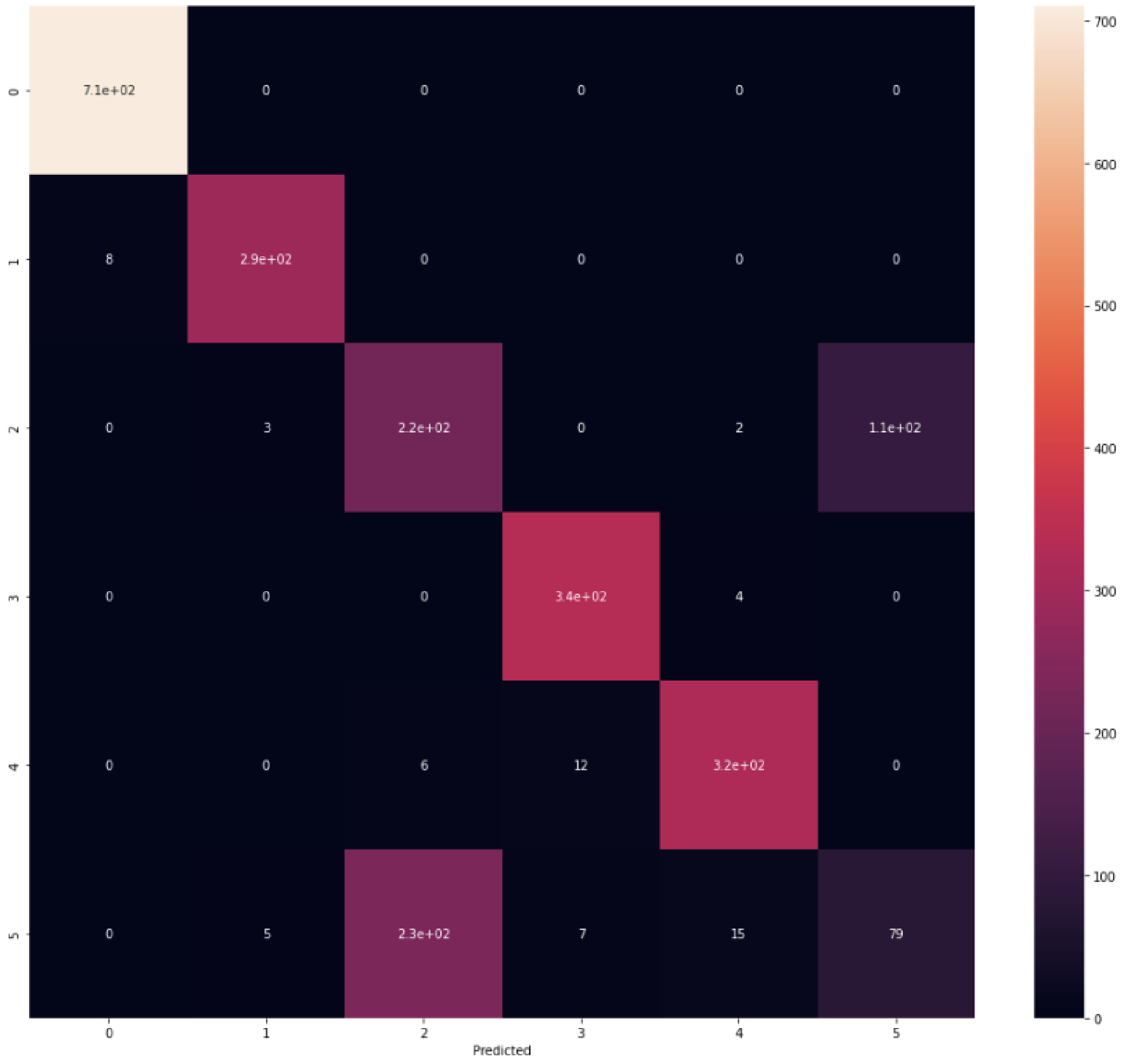
For KNN we first tested multiple values for n_neighbors resulting in our choice to use 6 neighbors. Again we used a stratified shuffle split with the same random state as in our LR models to ensure uniformity across both tests.



Looking at our classification report and our confusion matrix we can see that our model performed significantly better than base Logistic Regression. We have an f1-score greater than 0.90 on all faults with the exception of 0111 and 1111. These faults are both a three phase fault, the 1111 fault just also has a ground.

```
              precision    recall  f1-score   support

           0       0.99      1.00      0.99       710
         110       0.97      0.97      0.97       301
         111       0.47      0.66      0.55       329
        1001       0.95      0.99      0.97       339
        1011       0.94      0.95      0.94       340
        1111       0.42      0.23      0.30       340
```



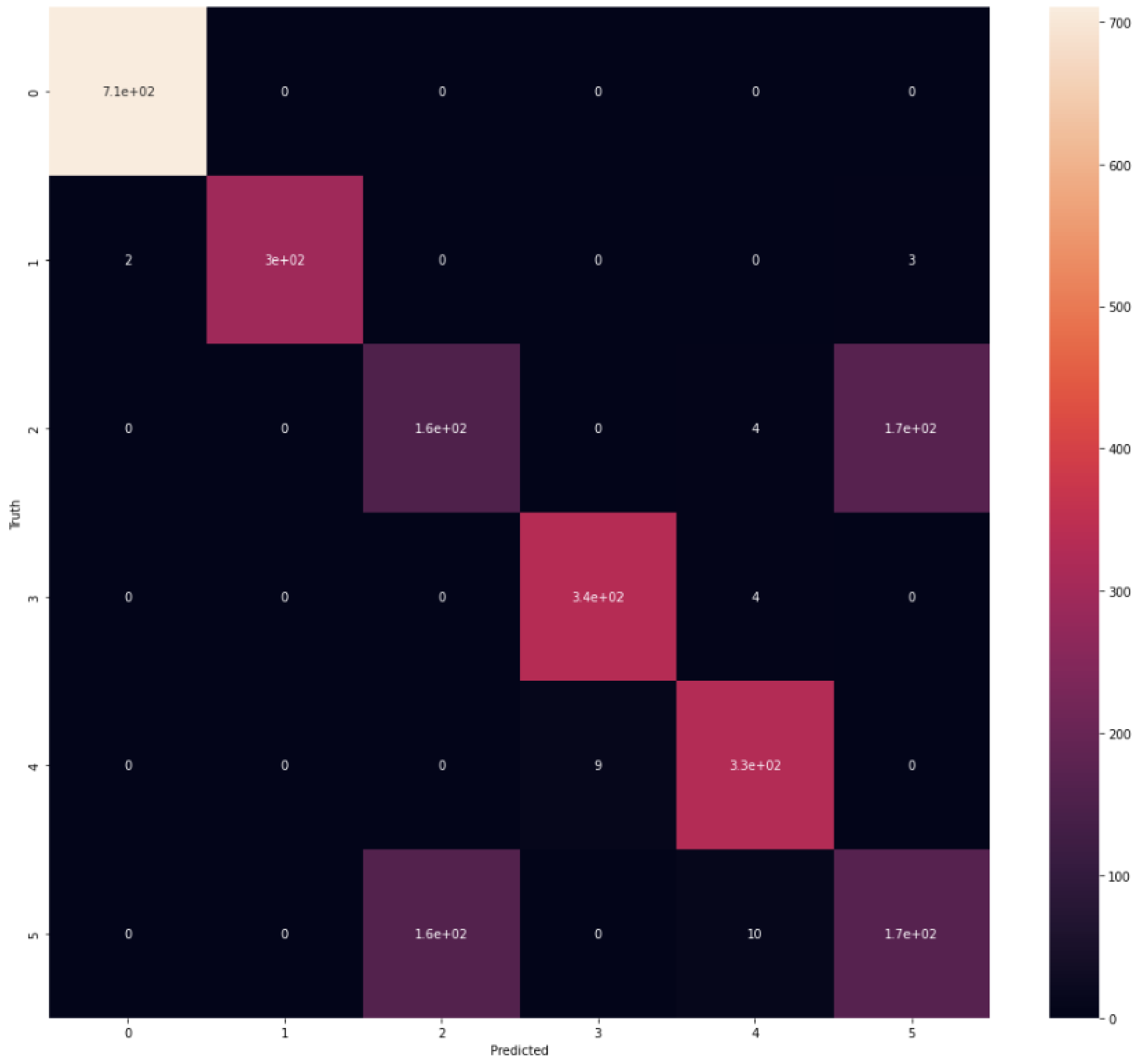
**SVC:**

Similar to KNN we tested multiple values for “C” landing on 15. We also tested multiple values for gamma and different kernels but the default’s of scale and RB performed the best.

Again we still run into the same problem as KNN where the model has difficulty telling the difference between fault 0111 and 1111 but it does do slightly better. Still not good enough to be satisfied with those two results though. Here is the classification report and the confusion matrix.

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       710
         110       1.00      0.98      0.99       301
         111       0.49      0.47      0.48       329
        1001       0.97      0.99      0.98       339
        1011       0.95      0.97      0.96       340
        1111       0.49      0.49      0.49       340
```



## A paragraph explaining which of your classifier models you recommend as a final model that best fits your needs in terms of accuracy and explainability.

I recommend the SVC model as the final model. Our goal for this project did not focus on explainability at all. We wanted a model that would have the best accuracy when predicting faulted conditions and that was achieved with the SVC model. We chose not to put any emphasis on explainability because this tool is intended to run in the background and provide recommendations or queues to the operator.

## Summary Key Findings and Insights, which walks your reader through the main drivers of your model and insights from your data derived from your classifier model.

Key Findings:

- Both KNN and SVC models performed extremely well.
- The models have a very hard time determining between a three phase fault, and a three phase fault with a ground.
- You would need a separate dataset and create a model for determining three phase vs three phase with ground faults. Or another feature specifically related to the line's relationship to ground.

Insights:

- I decided not to do a decision tree. I believe it would have gotten a better accuracy score but at the cost of a higher chance of overfitting was not worth the risk in this application.
- I decided to create the model based on the fault categories data instead of the binary fault condition to provide better insight for the company.

## Suggestions for next steps in analyzing this data, which may include suggesting revisiting this model after adding specific data features that may help you achieve a better explanation or a better prediction.

The only thing this data is missing is a way to check the ground condition. If that was a feature I think the accuracy of this model would be ~98%. I would like to revisit this at some point and make a separate model to only do the 3 phase vs 3 phase with ground faults.