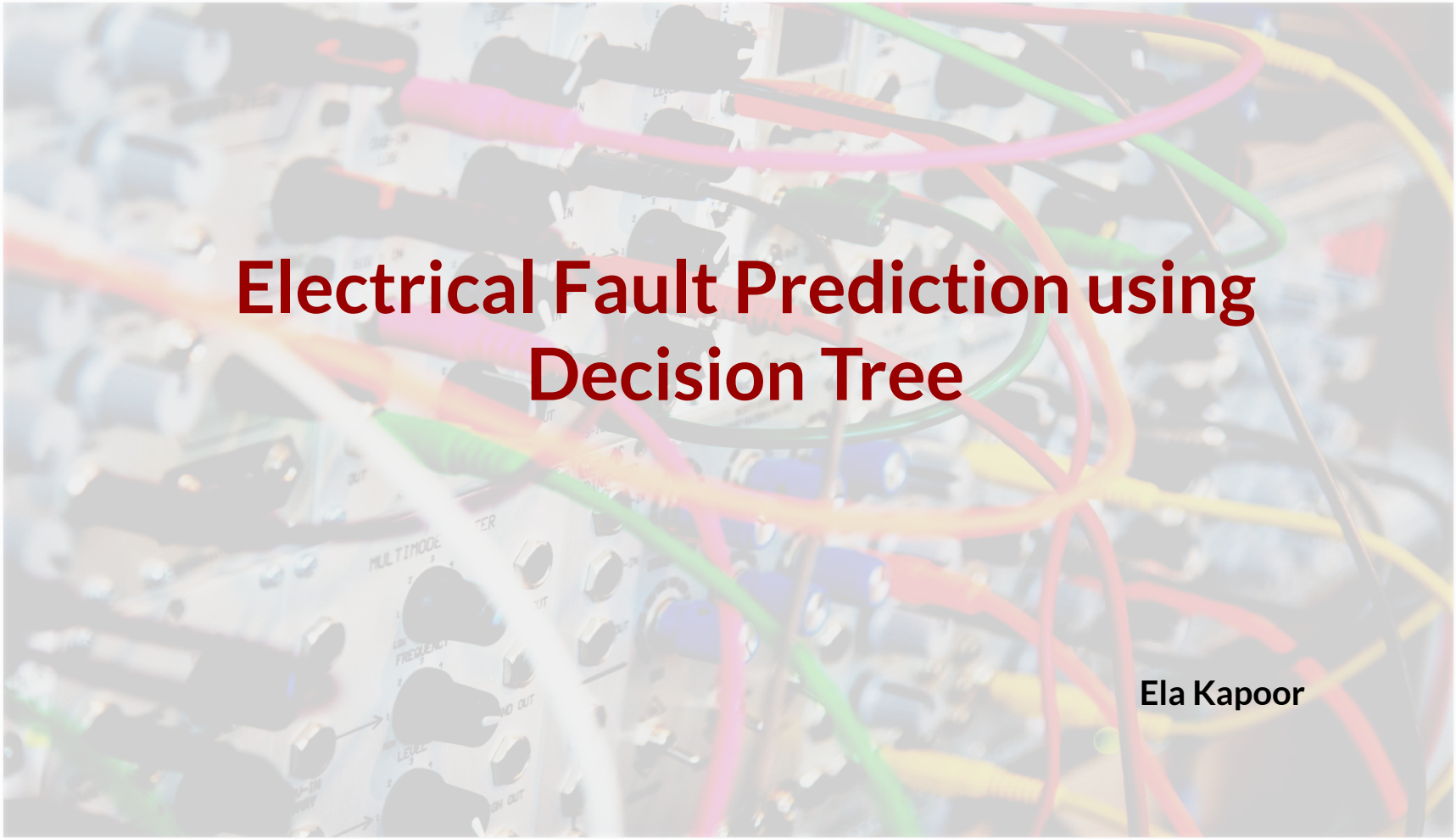


Electrical Fault Prediction using Decision Tree

Ela Kapoor

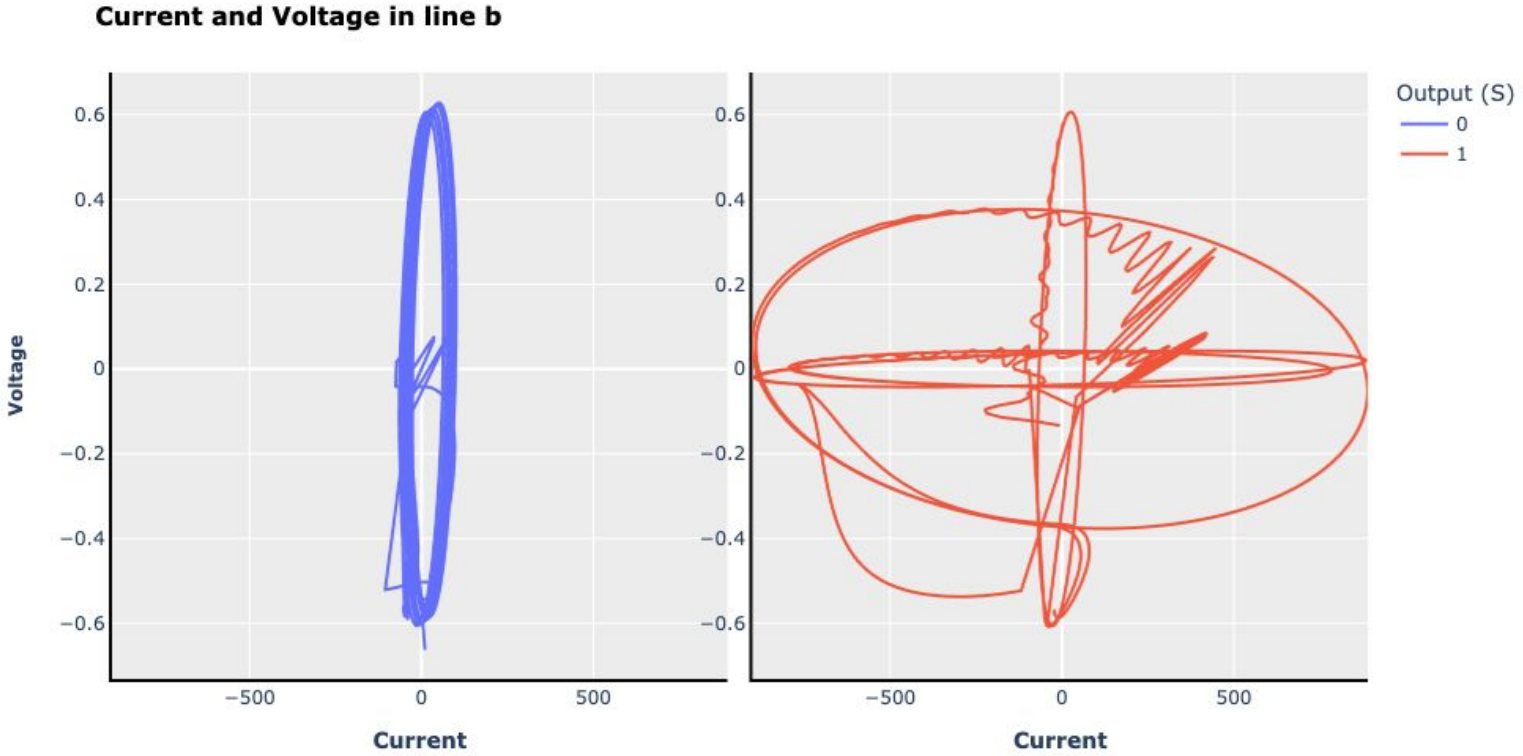
—

The electrical power system consists of many complex, dynamic and interacting elements that are always prone to disturbance or an electrical fault such as short circuit condition.

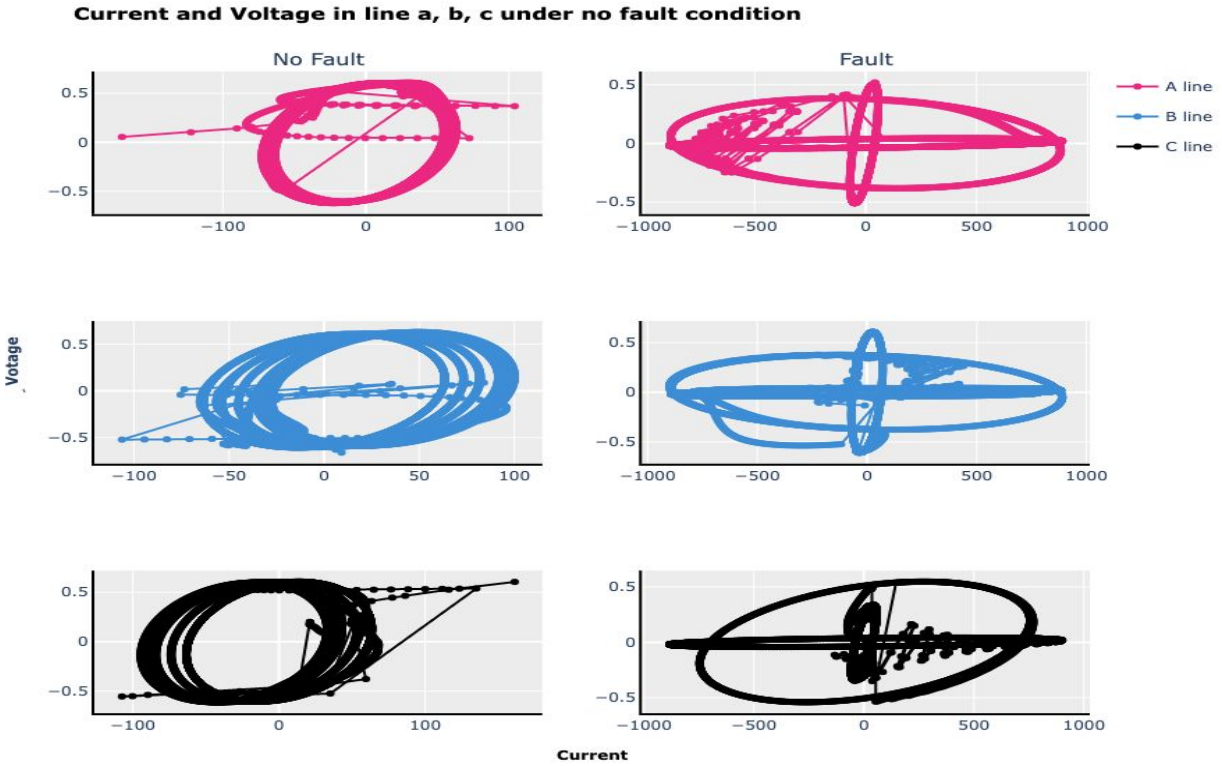
- Required fault detection system
- Operation of protection equipment in minimum possible time to remain stable.
- Initiate other relays to protect the power system from outages

— Output (label) is binary classified
  Fault ---> 1
  No Fault ---> 0
  Features are current and voltages in line a, b and c

	Output (S)	Ia	Ib	Ic	Va	Vb	Vc	Unnamed: 7	Unnamed: 8
6487	0	10.009379	-43.194571	35.379810	0.597965	-0.275271	-0.322694	NaN	NaN
7445	1	73.138358	-798.340255	727.203438	-0.035802	-0.001706	0.037508	NaN	NaN
1705	0	43.220846	-65.293233	29.318940	0.580671	-0.123633	-0.457038	NaN	NaN
440	0	-29.728845	-33.659446	63.388292	0.462295	-0.570358	0.108063	NaN	NaN
6706	1	765.982618	-772.398070	8.564309	-0.001782	-0.035833	0.037614	NaN	NaN

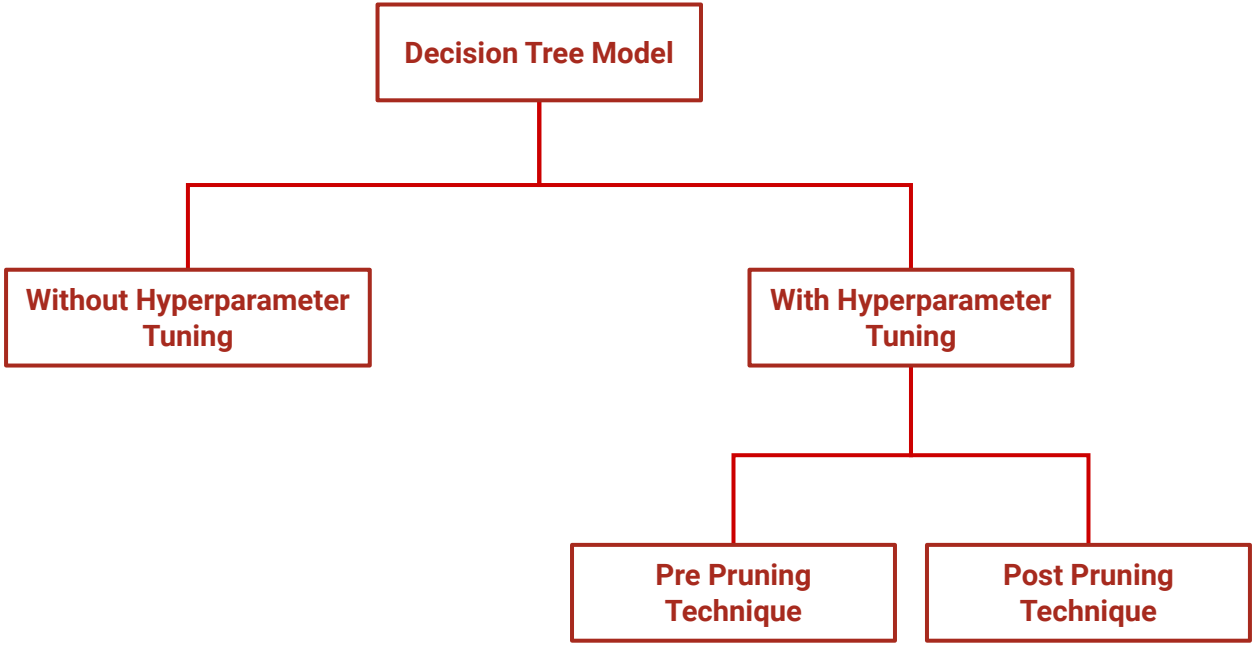


PROBLEM STATEMENT | EDA | MODEL DESIGN | CONCLUSION



PROBLEM STATEMENT | EDA | MODEL DESIGN | CONCLUSION

Correlation Heatmap


Decision Tree Model
Without Hyperparameter Tuning
With Hyperparameter Tuning
Pre Pruning Technique
Post Pruning Technique

MODEL ACCURACY

Without Tuning

Test Accuracy → 99.44%
Train Accuracy → 100%

- No hyper parameter tuning

Pre-Pruning

Test Accuracy → 99.38%
Train Accuracy → 99.91%

- max_depth
- min_samples_leaf

Post-Pruning

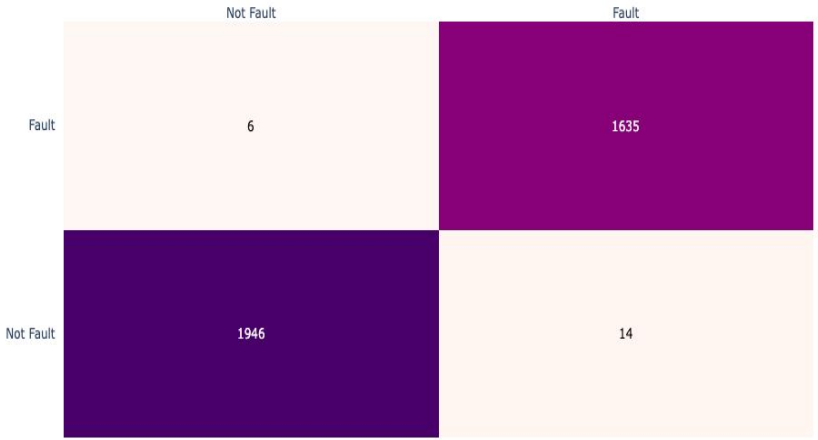
Test Accuracy → 99.30%
Train Accuracy → 99.35%

- Changing the value of alpha

Without Tuning

Train Data

Test Data



Pre Pruning Techniques

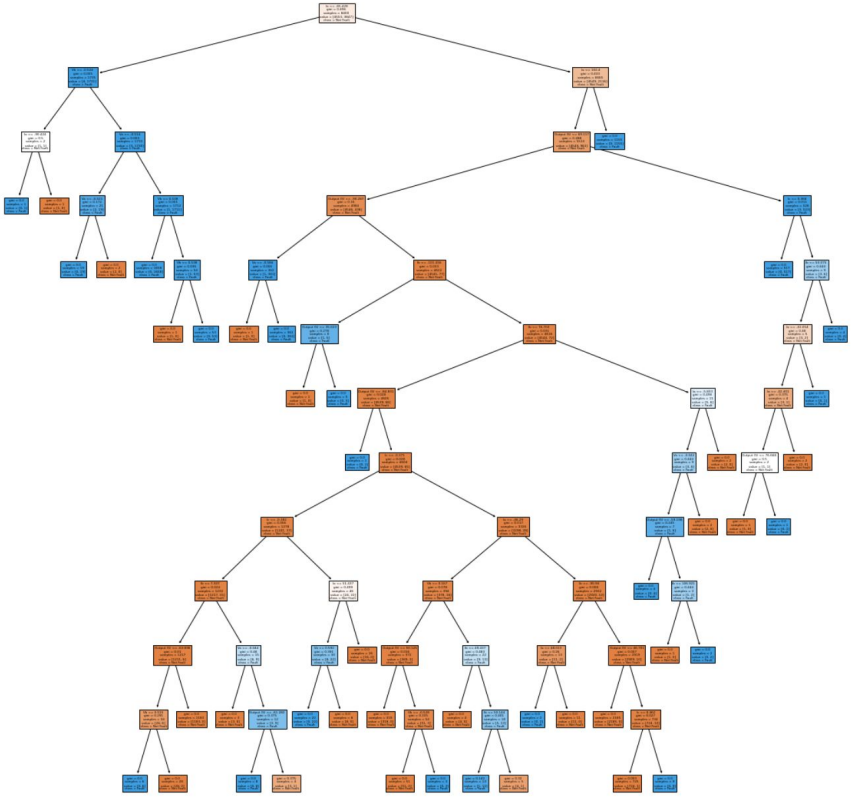
Train Data

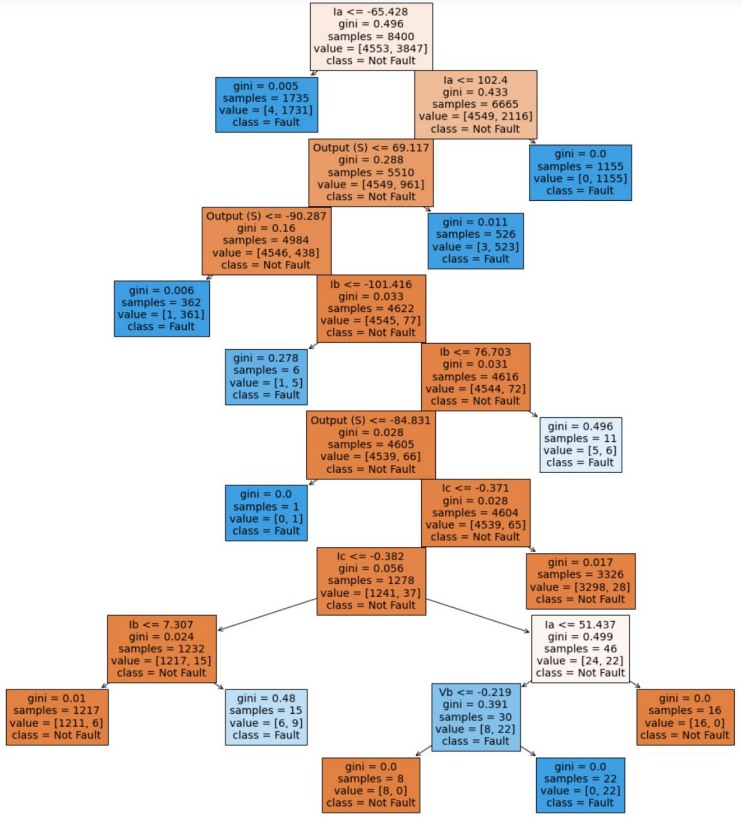
Test Data

Post Pruning Techniques

Train Data

Test Data




Ia <= -65.428
gini = 0.496
samples = 8400
value = [4553, 3847]
class = Not Fault
gini = 0.005
samples = 1735
value = [4, 1731]
class = Fault
Ia <= 102.4
gini = 0.433
samples = 6665
value = [4549, 2116]
class = Not Fault
Output (S) <= 69.117
gini = 0.288
samples = 5510
value = [4549, 961]
class = Not Fault
gini = 0.0
samples = 1155
value = [0, 1155]
class = Fault
Output (S) <= -90.287
gini = 0.16
samples = 4984
value = [4546, 438]
class = Not Fault
gini = 0.011
samples = 526
value = [3, 523]
class = Fault
gini = 0.006
samples = 362
value = [1, 361]
class = Fault
Ib <= -101.416
gini = 0.033
samples = 4622
value = [4545, 77]
class = Not Fault
gini = 0.278
samples = 6
value = [1, 5]
class = Fault
Ib <= 76.703
gini = 0.031
samples = 4616
value = [4544, 72]
class = Not Fault
Output (S) <= -84.831
gini = 0.028
samples = 4605
value = [4539, 66]
class = Not Fault
gini = 0.496
samples = 11
value = [5, 6]
class = Fault
gini = 0.0
samples = 1
value = [0, 1]
class = Fault
Ic <= -0.371
gini = 0.028
samples = 4604
value = [4539, 65]
class = Not Fault
Ic <= -0.382
gini = 0.056
samples = 1278
value = [1241, 37]
class = Not Fault
gini = 0.017
samples = 3326
value = [3298, 28]
class = Not Fault
Ib <= 7.307
gini = 0.024
samples = 1232
value = [1217, 15]
class = Not Fault
Ia <= 51.437
gini = 0.499
samples = 46
value = [24, 22]
class = Not Fault
gini = 0.01
samples = 1217
value = [1211, 6]
class = Not Fault
gini = 0.48
samples = 15
value = [6, 9]
class = Fault
Vb <= -0.219
gini = 0.391
samples = 30
value = [8, 22]
class = Fault
gini = 0.0
samples = 16
value = [16, 0]
class = Not Fault
gini = 0.0
samples = 8
value = [8, 0]
class = Not Fault
gini = 0.0
samples = 22
value = [0, 22]
class = Fault

- We can see that the difference between the accuracy on the train set and test set decreased. This is because hyperparameter tuning smoothens the decision boundary and thus prevents it from overfitting.

- The model accuracy is good and can be implemented for production environment.

- Following benefits because of model:
  - Reduce the frequency of maintenance
  - Minimizes cost of maintenance
  - Save life
  - Avoid and minimize downtime
  - Increase availability of the system

Thank You!

Wishing you
Happy Autumn