PiWIN

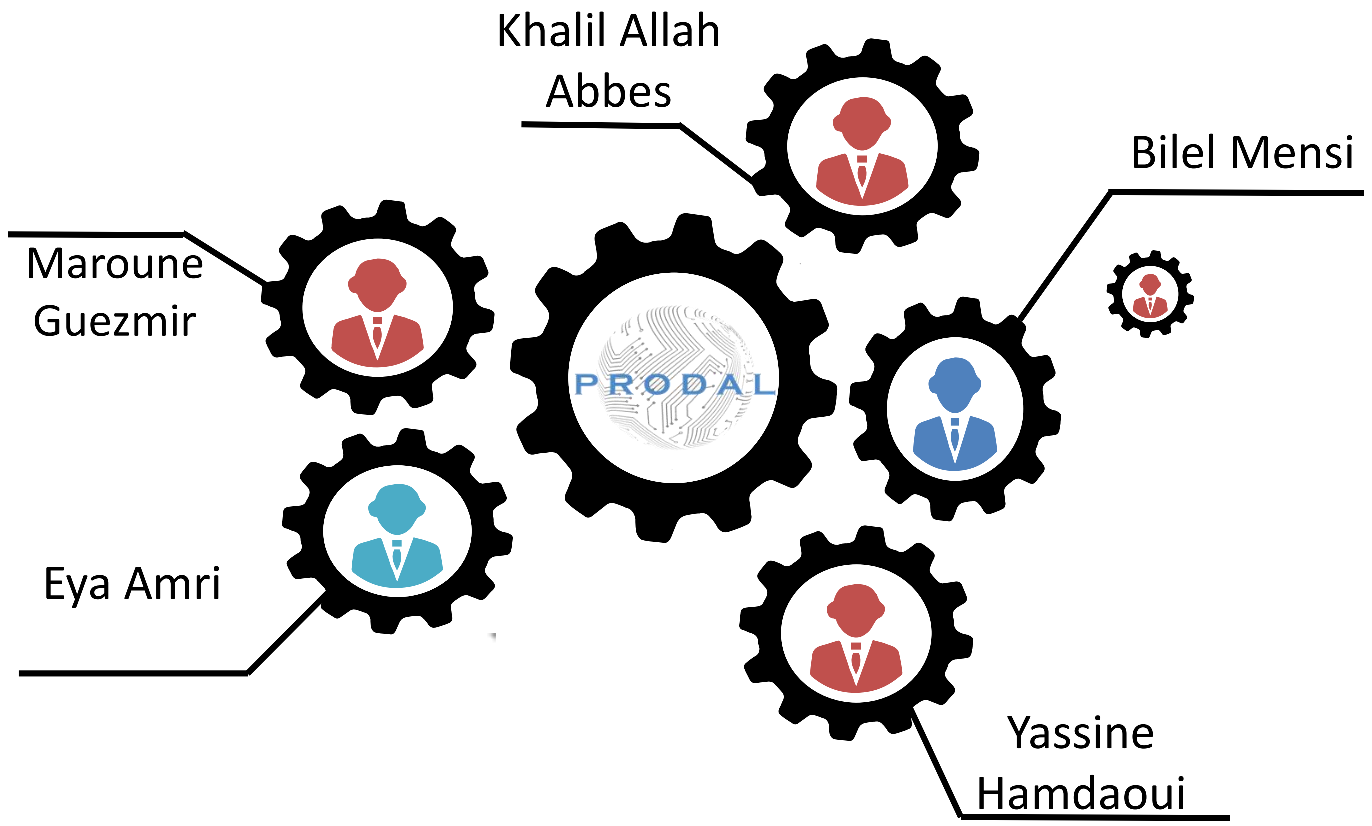
Radio Link Failure Prediction Project

Realized by: P R O D A L

Phase III

Summary:

The Team:

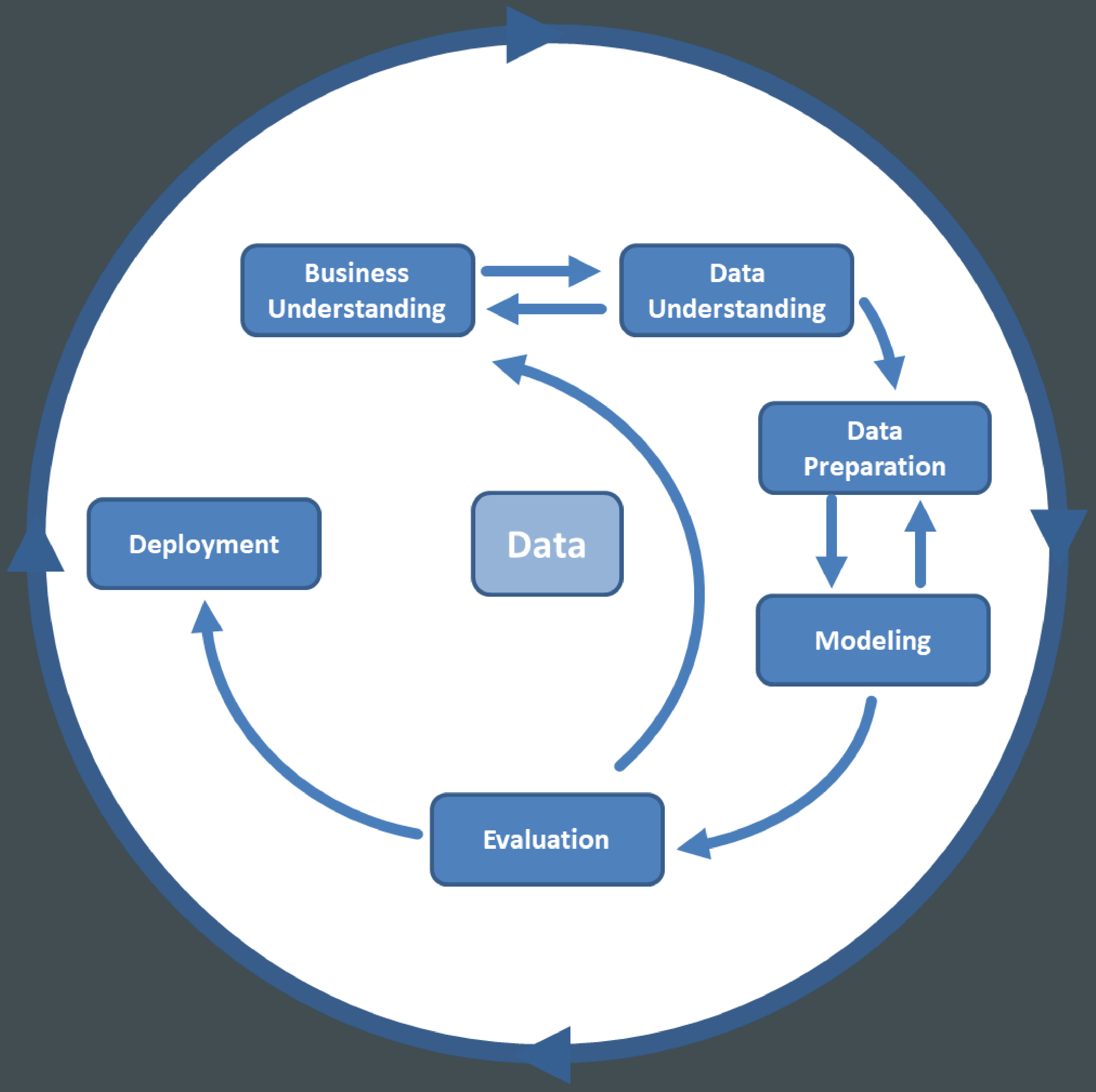


**ALONE WE CAN DO SO LITTLE,
TOGETHER WE CAN DO SO MUCH
-HELLEN KELLER**

Introduction

The goal of our project is to limit the radio link failure by exploiting the data to guarantee an ideal environment.

To ensure an ideal transmission environment we will implement several models to predict radio link failures.



Business objectives:

- **Ensure performance of the network by avoiding network failures**
- **Reduce the effect of the weather on the performance of the radio link**
- **Financial gain and customer satisfaction**

Data science objectives:

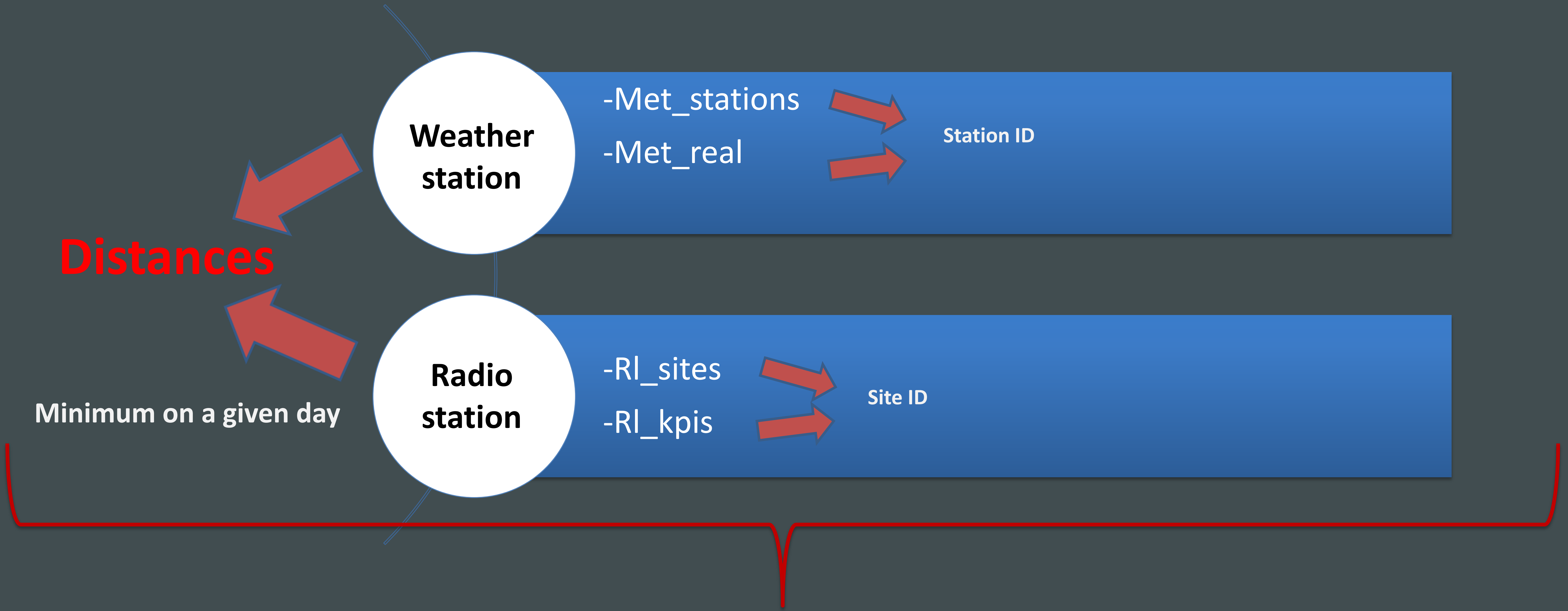
- Predict radio link failure for next day and next 5 days based on region, historical data and weather data.
- Implement a prediction system to identify the causes of the radio link failure
- Identify weather conditions that affect directly the network
- Anticipate the settings to be made on the radio stations to avoid radio link failure

Data preparation:

Code:

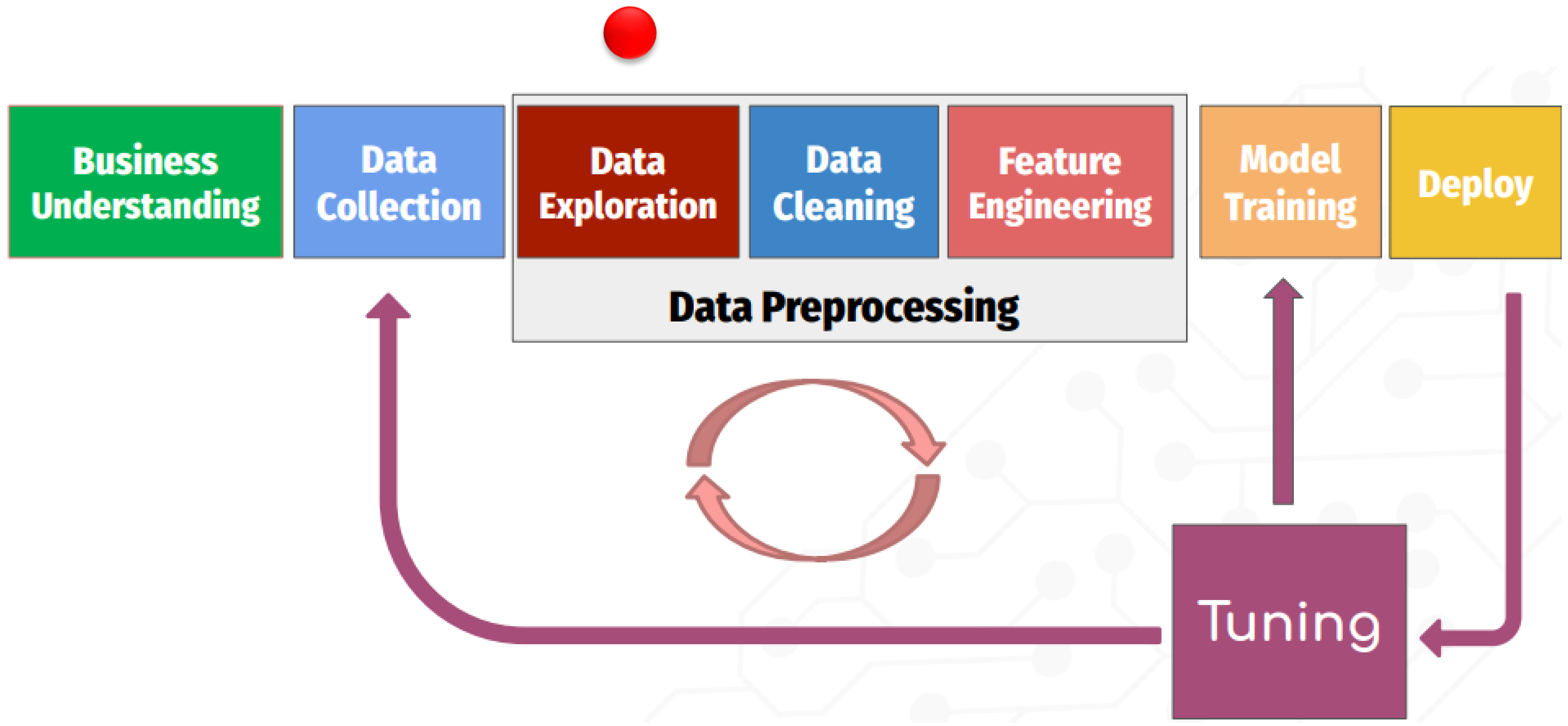
```
xlsx = pd.ExcelFile('data-Test-2020-09-01.xlsx')
sheets = pd.read_excel(xlsx, sheet_name=None, index_col=0,
                       na_filter=True, convert_float=False)

met_stations = sheets['met-stations']
met_real = sheets['met-real']
met_forecast = sheets['met-forecast']
rl_sites = sheets['rl-sites']
rl_kpis = sheets['rl-kpis']
distances = sheets['distances']

# Station names
stations = met_stations['station_no'].tolist()

# Forecast dates
forecast_dates = sorted(list(set(met_forecast['datetime'])))
```

Merged dataset:



Filtred by Met_forecast

Data exploration:

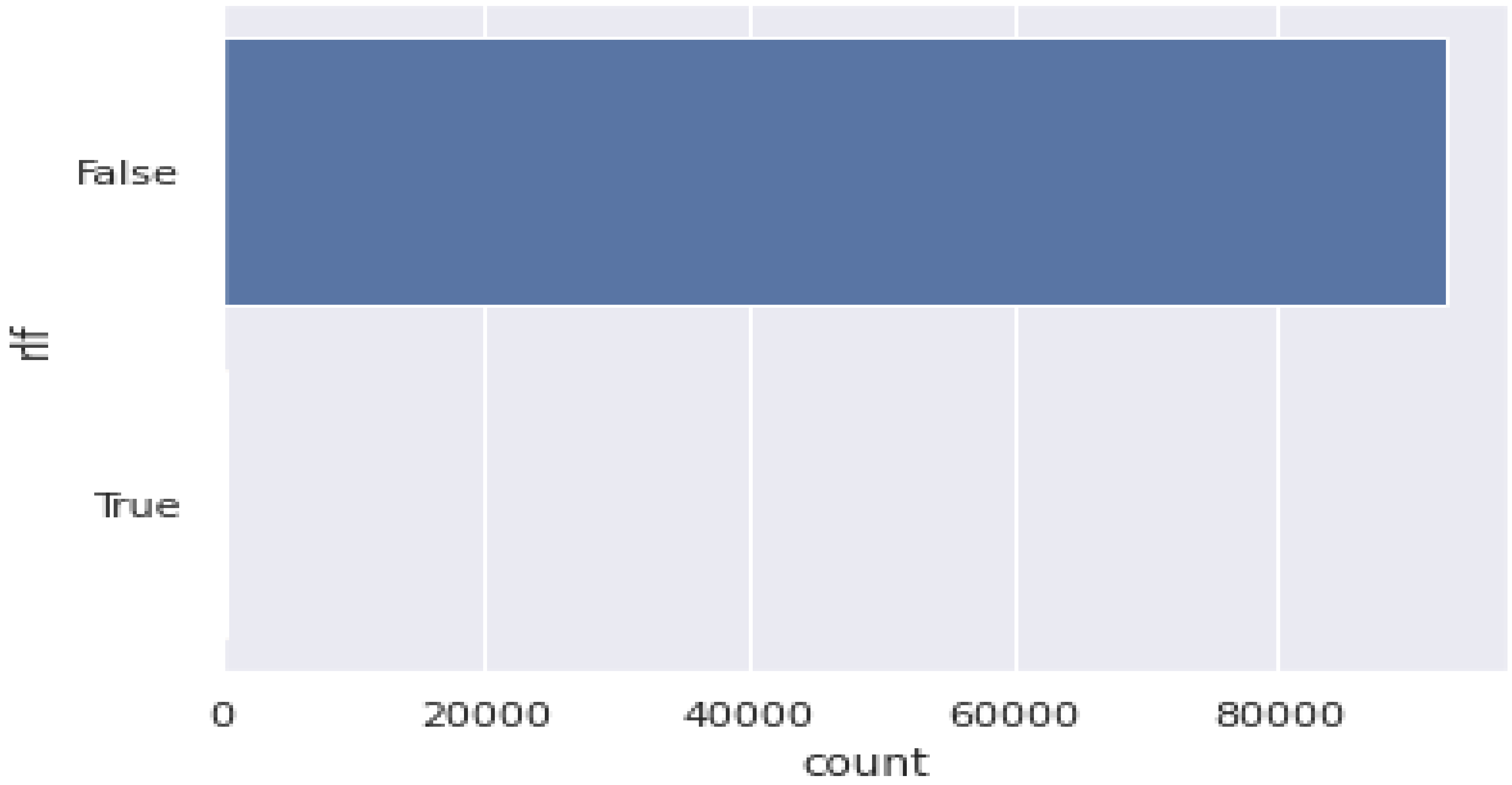


Statistical Description(93250 * 67) :

	mw_connection_no	site_no	link_length	severaly_error_second	error_second	unavail_second	avail_time	bbe
count	9.325000e+04	93250.000000	14606.000000	93250.000000	93250.000000	93250.000000	93250.000000	93250.000000
mean	4.796969e+05	37249.962155	5958.605847	0.160097	1.934273	305.722177	84541.888665	67.034198
std	4.509775e+05	31923.074906	5854.243154	20.688216	43.597285	4888.954324	10784.345415	3358.574491
min	1.435940e+05	277.000000	35.000000	0.000000	0.000000	0.000000	1.000000	0.000000
25%	2.088880e+05	10404.000000	1333.000000	0.000000	0.000000	0.000000	86400.000000	0.000000
50%	2.959210e+05	23989.000000	3500.000000	0.000000	0.000000	0.000000	86400.000000	0.000000
75%	3.514740e+05	62702.000000	10269.000000	0.000000	0.000000	0.000000	86400.000000	0.000000
max	1.384577e+06	99297.000000	30279.000000	6238.000000	7191.000000	86400.000000	108000.000000	600513.000000

8 rows × 40 columns

Classes Distribution:




```
False    92825
True       425
Name: rlf, dtype: int64
```

Null Values (13 columns):

```
polarization                79847
freq_band                     376
link_length                 78644
scalibility_score           48093
humidity_max_day1             602
humidity_min_day1             602
wind_dir_day1                 602
wind_speed_day1               602
history_polarization        80447
history_freq_band             376
history_link_length         79245
history_scalibility_score   49865
history_clutter_class         393
dtype: int64
```

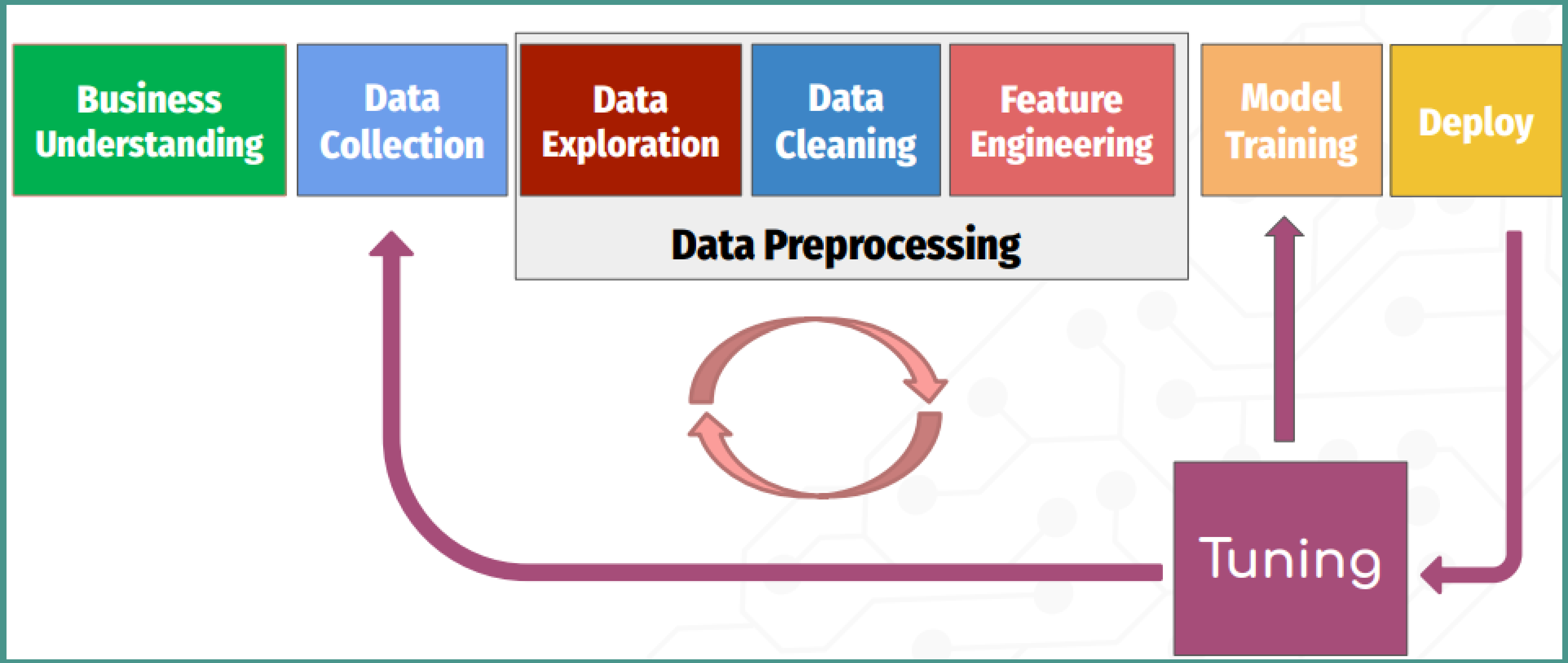
Descision Making

< 1000 - imputer

> 1000 : Delete

- Heavily imbalanced class labels (0,06% « True » labels)
- Uneven Data samples
- Insufficient converge of all possible radio link failures scenarios
- Missing values (18%)
- Unavailibity of actual weather forecast from RL sites,

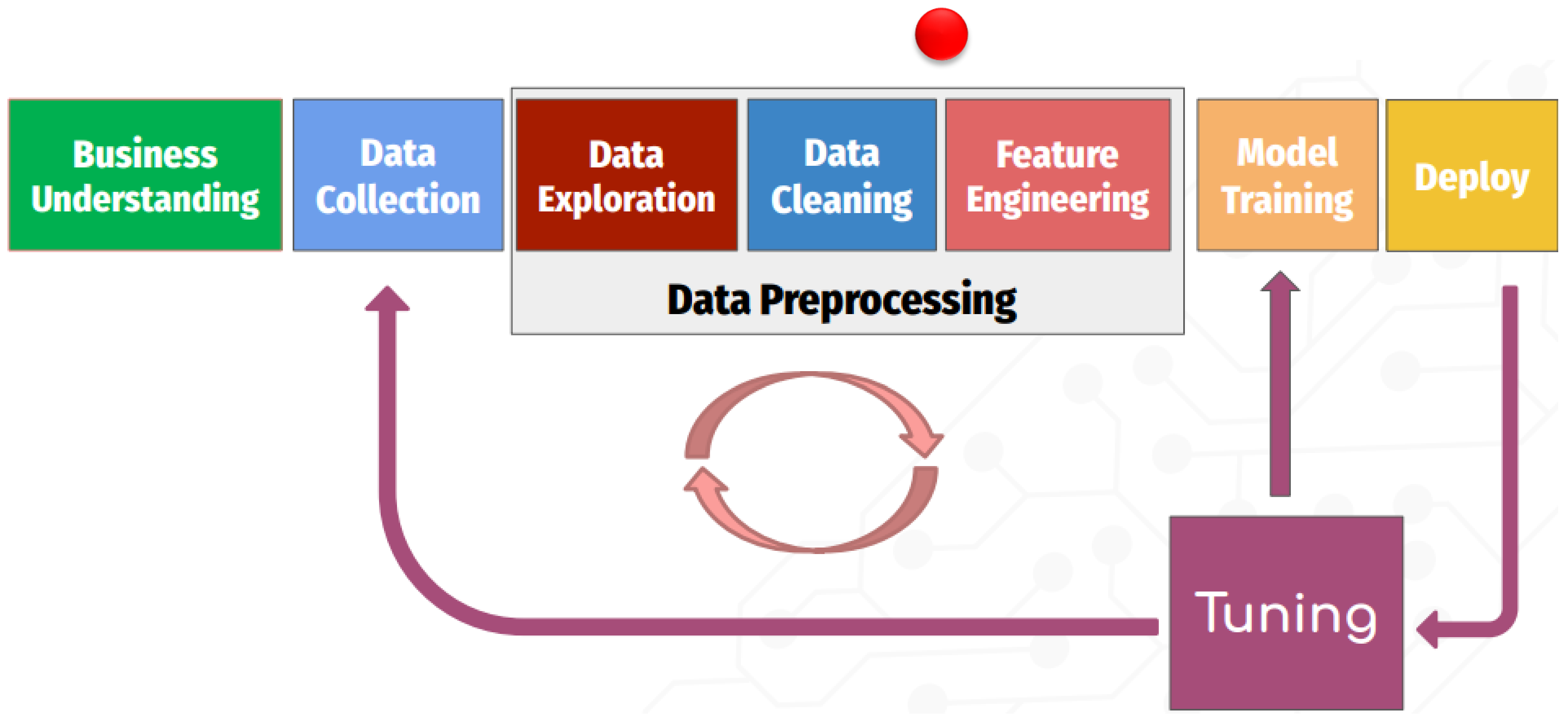
Data cleaning



On Merged dataset

- Dropping some links according to outliers RF Links analysis based on triangulation.
- Dropping some NA values
- Upsampling and downsampling based on the month

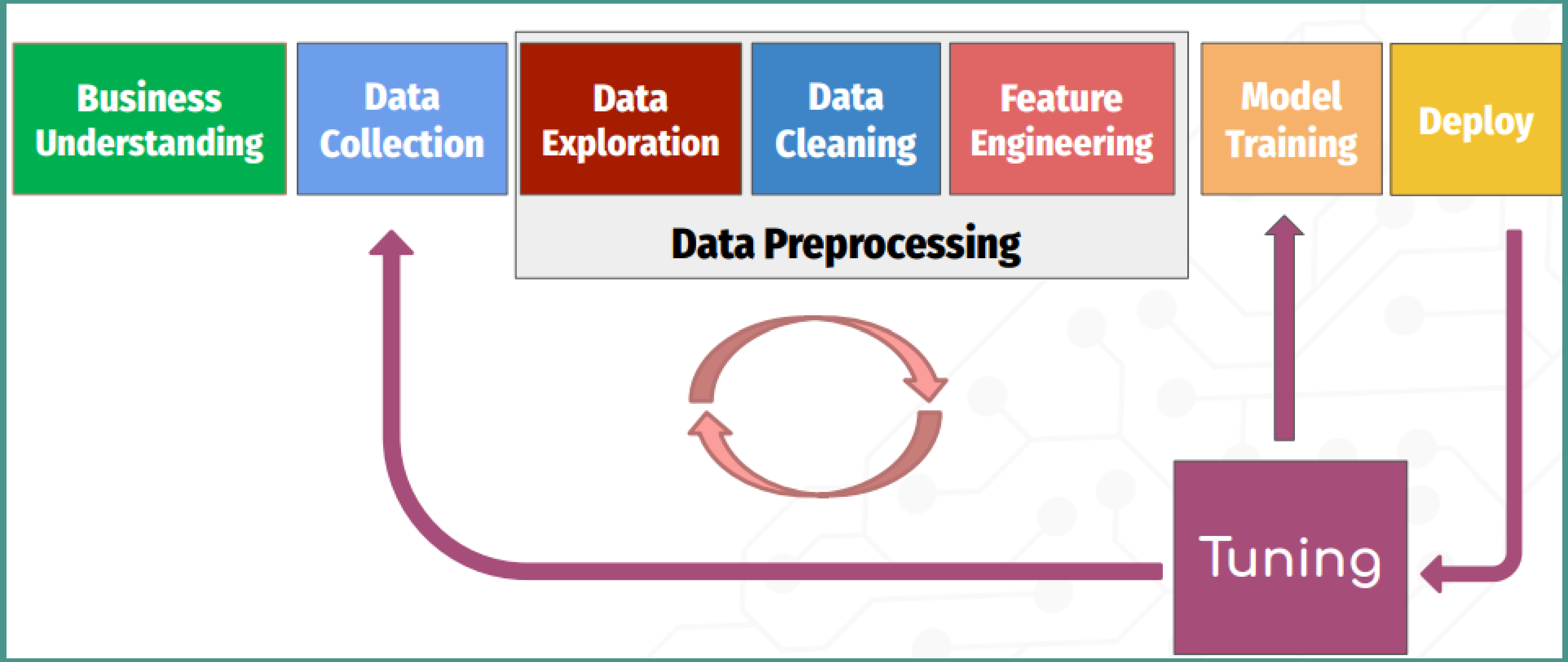
Data transformation:



- **We have a regression model so we decided to Normalize numerical columns such as “bbe” and “unvail_seconds” by using standard scalling**

- **We have a binary classifier so we decided to use One Hot Encoding for categorical columns such as “history_polarization” and “adaptive_modulation”**

Feature engineering:

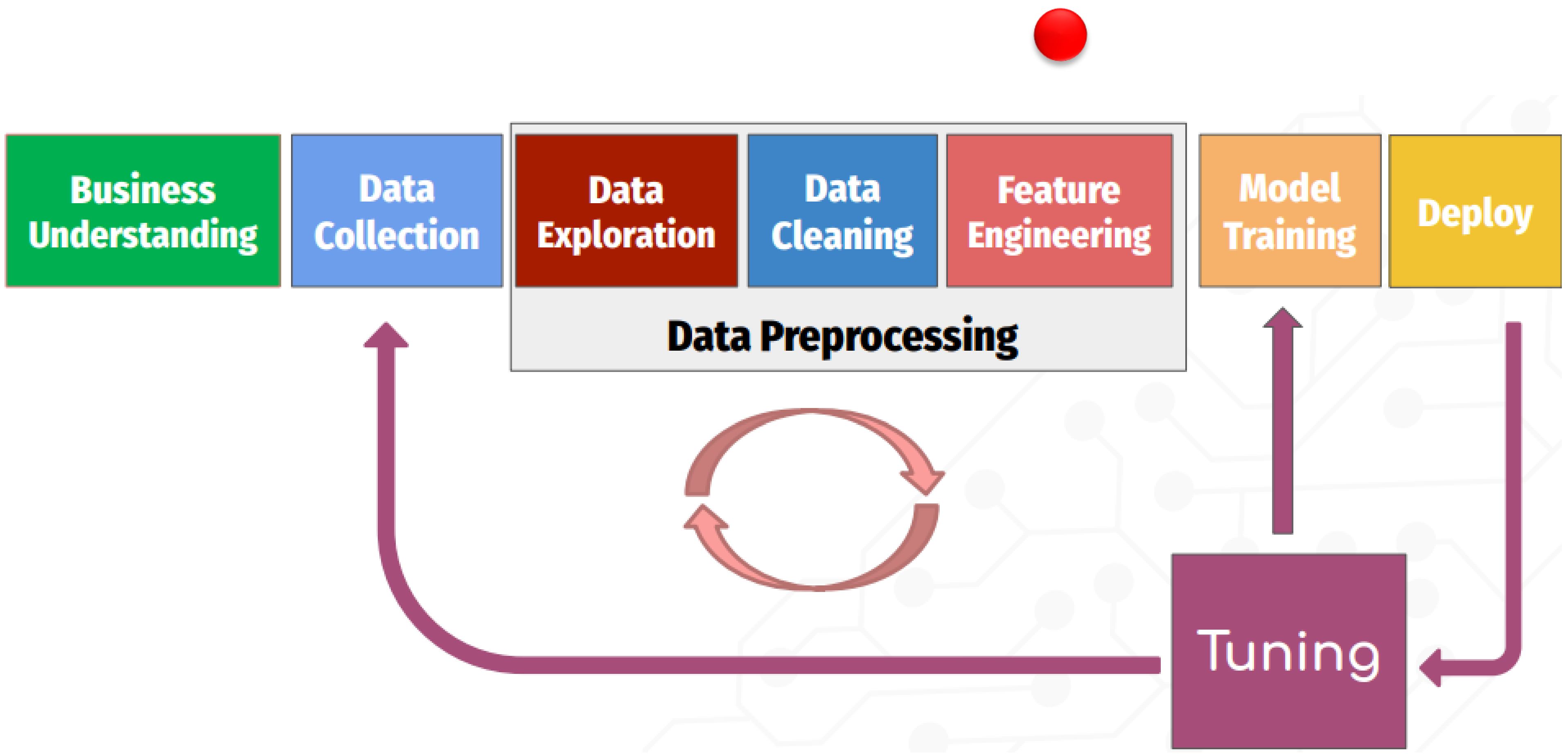


- **Date/Time feature** (Year / Month / Day)

- **Aggregation feature** (sum/count/max/min)

- **Polynomial** feature

Feature selection:



- **Select feature by importance**

- **Select feature by k-best**

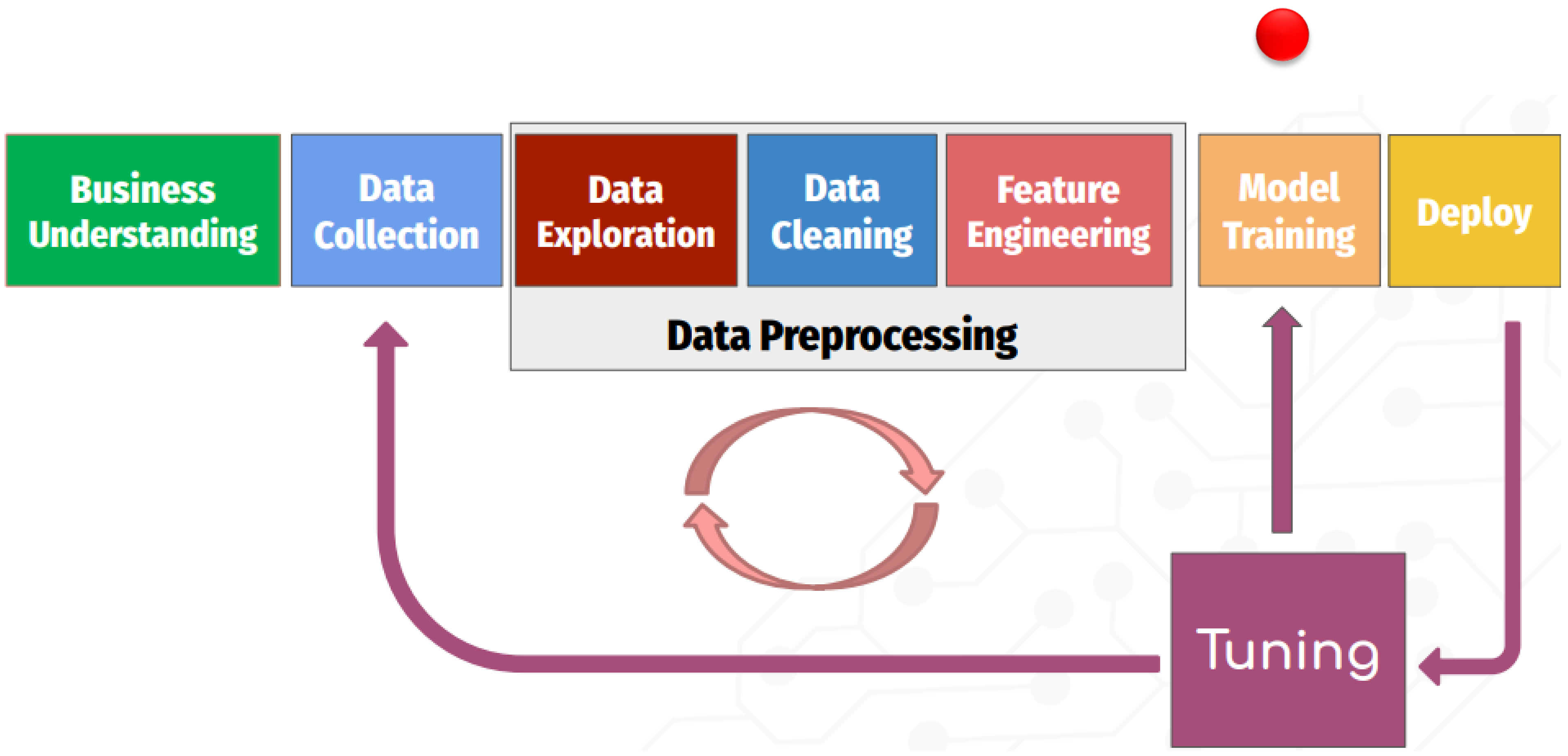
- **Select feature by k-square**

- **Select feature by k-scoring**

	rxlevmax	capacity	temp_max_day1	temp_min_day1	humidity_max_day1	humidity_min_day1	wind_dir_day1	wind_speed_day1	wd1_few clouds	wd1_heav ra
19800	-28.2	456.0	14.0	5.0	71	43	349	15	0	
19802	-39.5	160.0	14.0	5.0	71	43	349	15	0	
19872	-30.0	456.0	14.0	5.0	71	43	349	15	0	
19904	-33.3	247.0	14.0	5.0	71	43	349	15	0	
19953	-22.5	200.0	17.0	6.0	82	39	310	11	0	
...	...	...	...	...	...	...	...	...	...	
22405	-39.8	417.0	15.0	7.0	78	52	61	13	0	
22419	-37.6	74.0	15.0	7.0	78	52	61	13	0	
22431	-29.4	154.0	15.0	7.0	78	52	61	13	0	
22493	-39.8	456.0	7.0	2.0	92	77	151	4	0	
22552	-40.4	72.0	10.0	1.0	91	71	248	7	0	

117 rows × 30 columns

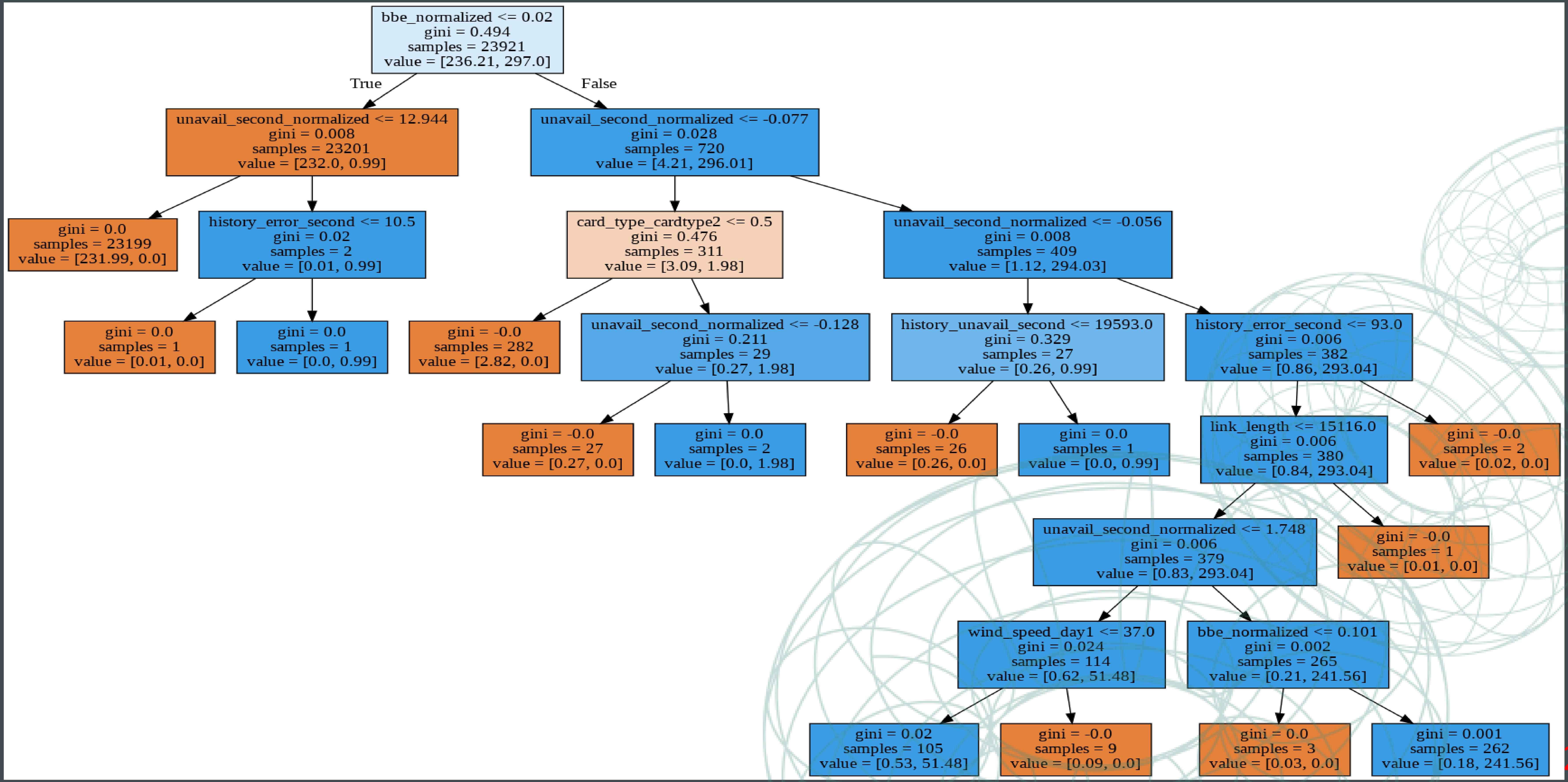
Modeling

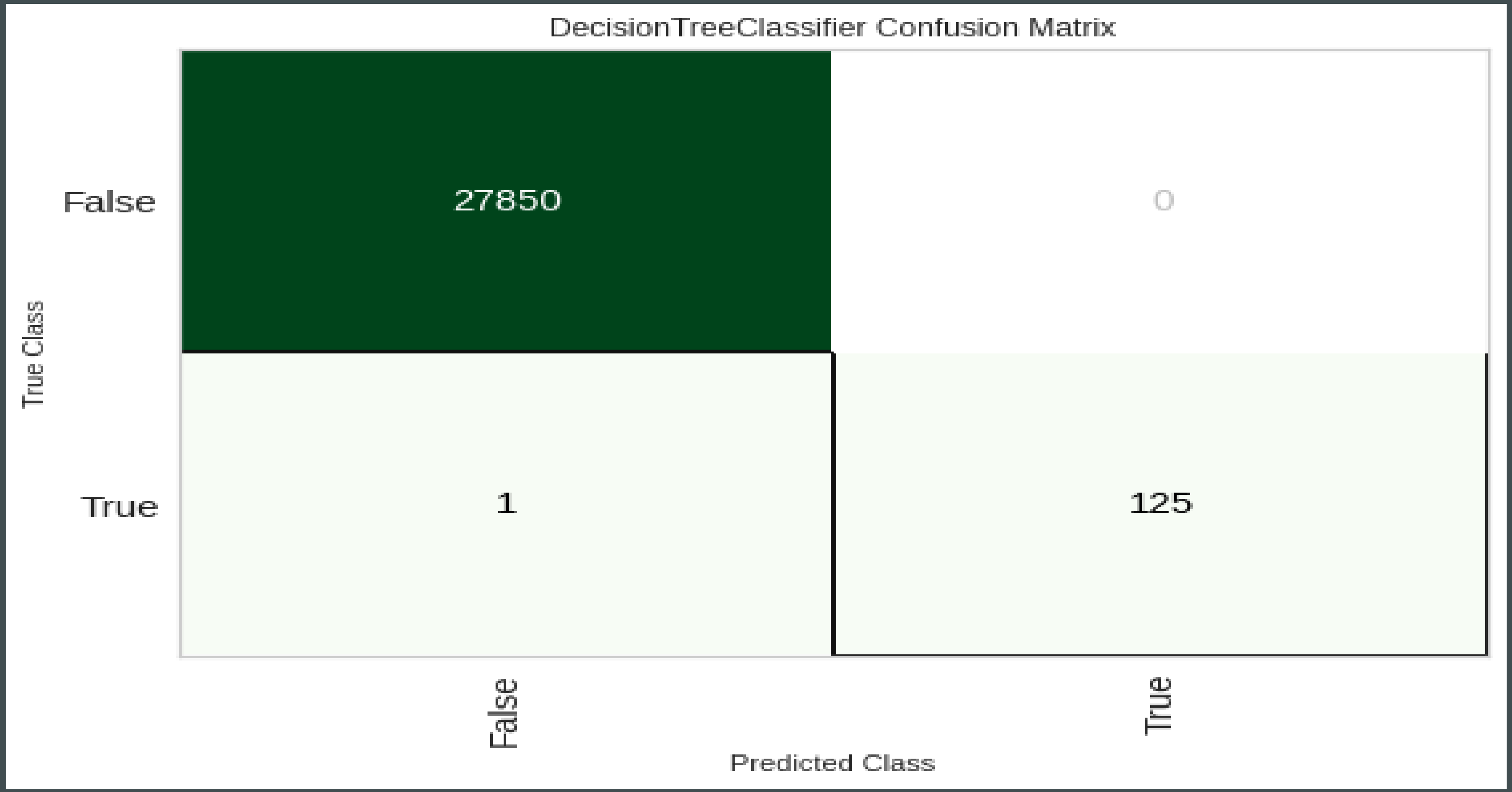

Business Understanding
Data Collection
Data Exploration
Data Cleaning
Feature Engineering
Model Training
Deploy
Data Preprocessing
Tuning

Camparing Models :

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
dt	Decision Tree Classifier	0.9672	0.9671	0.9950	0.9429	0.9682	0.9343	0.9360	0.0330
xgboost	Extreme Gradient Boosting	0.9922	0.9962	0.9950	0.9896	0.9923	0.9845	0.9845	1.3340
lightgbm	Light Gradient Boosting Machine	0.9920	0.9966	0.9945	0.9896	0.9920	0.9840	0.9840	0.5150
rf	Random Forest Classifier	0.9930	0.9976	0.9940	0.9920	0.9930	0.9860	0.9860	0.4680
et	Extra Trees Classifier	0.9807	0.9974	0.9860	0.9757	0.9808	0.9614	0.9615	0.3680
catboost	CatBoost Classifier	0.9779	0.9954	0.9679	0.9878	0.9777	0.9559	0.9562	3.8260
knn	K Neighbors Classifier	0.7774	0.8639	0.8902	0.7267	0.8000	0.5547	0.5698	0.0570
gbc	Gradient Boosting Classifier	0.9321	0.9654	0.8772	0.9855	0.9280	0.8641	0.8697	0.4120
ada	Ada Boost Classifier	0.8704	0.9242	0.8642	0.8753	0.8696	0.7408	0.7410	0.1860
lr	Logistic Regression	0.8368	0.8717	0.8296	0.8422	0.8355	0.6736	0.6742	1.2070
ridge	Ridge Classifier	0.8315	0.0000	0.8156	0.8427	0.8286	0.6631	0.6638	0.0480
lda	Linear Discriminant Analysis	0.8308	0.8733	0.8146	0.8421	0.8278	0.6616	0.6624	0.0630
svm	SVM - Linear Kernel	0.5811	0.0000	0.5990	0.6879	0.5451	0.1624	0.2147	0.1710
nb	Naive Bayes	0.6678	0.7380	0.5644	0.7131	0.6288	0.3356	0.3440	0.0230
qda	Quadratic Discriminant Analysis	0.5523	0.6623	0.4519	0.6776	0.4253	0.1043	0.1418	0.0390

Decision tree





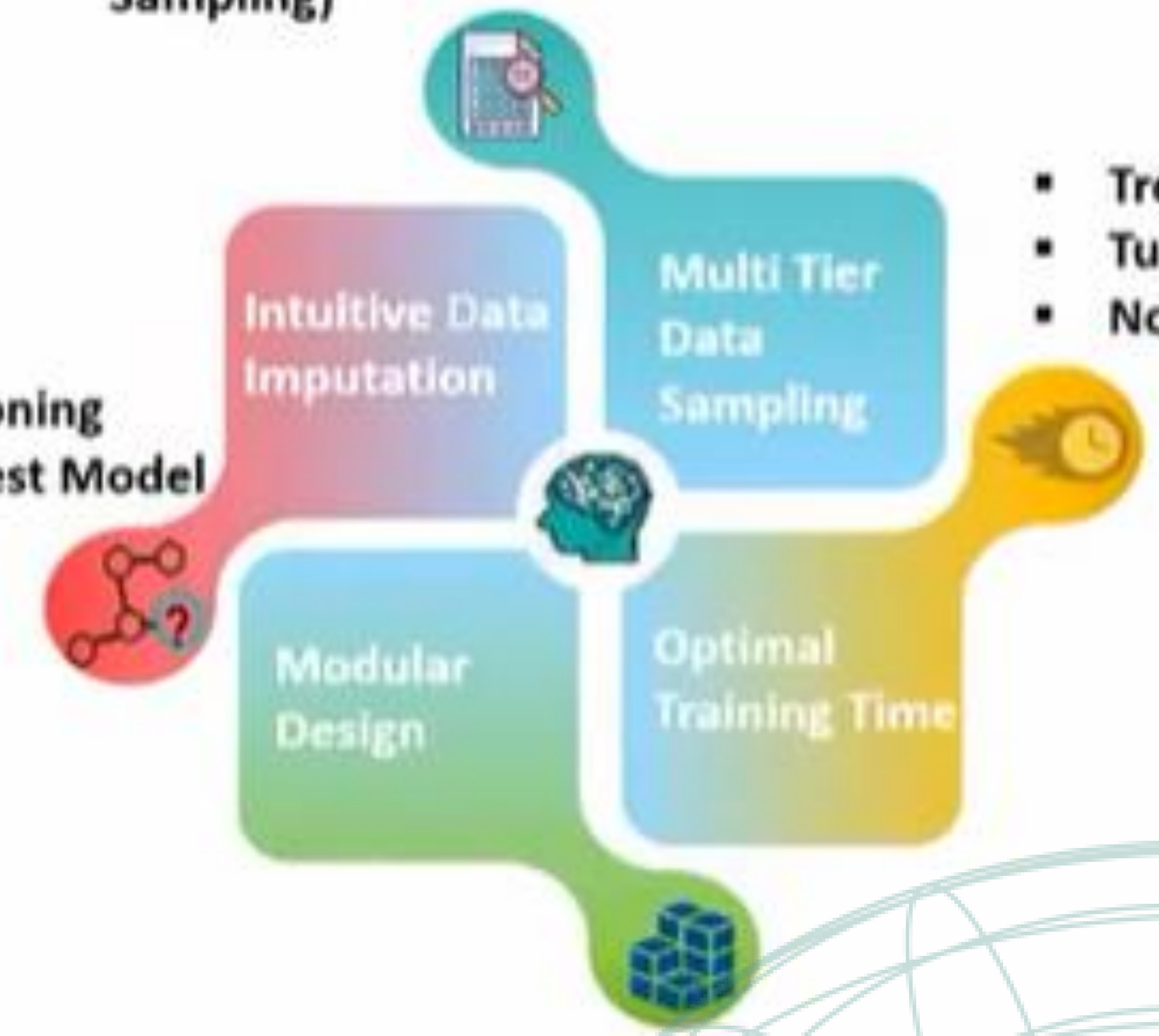
- **Tree based Algo**
- **Tuned model**
- **Normalised dataset**

Decision tree is the **BEST** model

Best Model Results	
F1-Score	99.74%
Accuracy	99.7%
Model	Extra Tree

- Mahanalobis
- SMOTE (over Sampling + under Sampling)

3 Fold Approach
- Logical Reasoning
- Random Forest Model
- K-NN

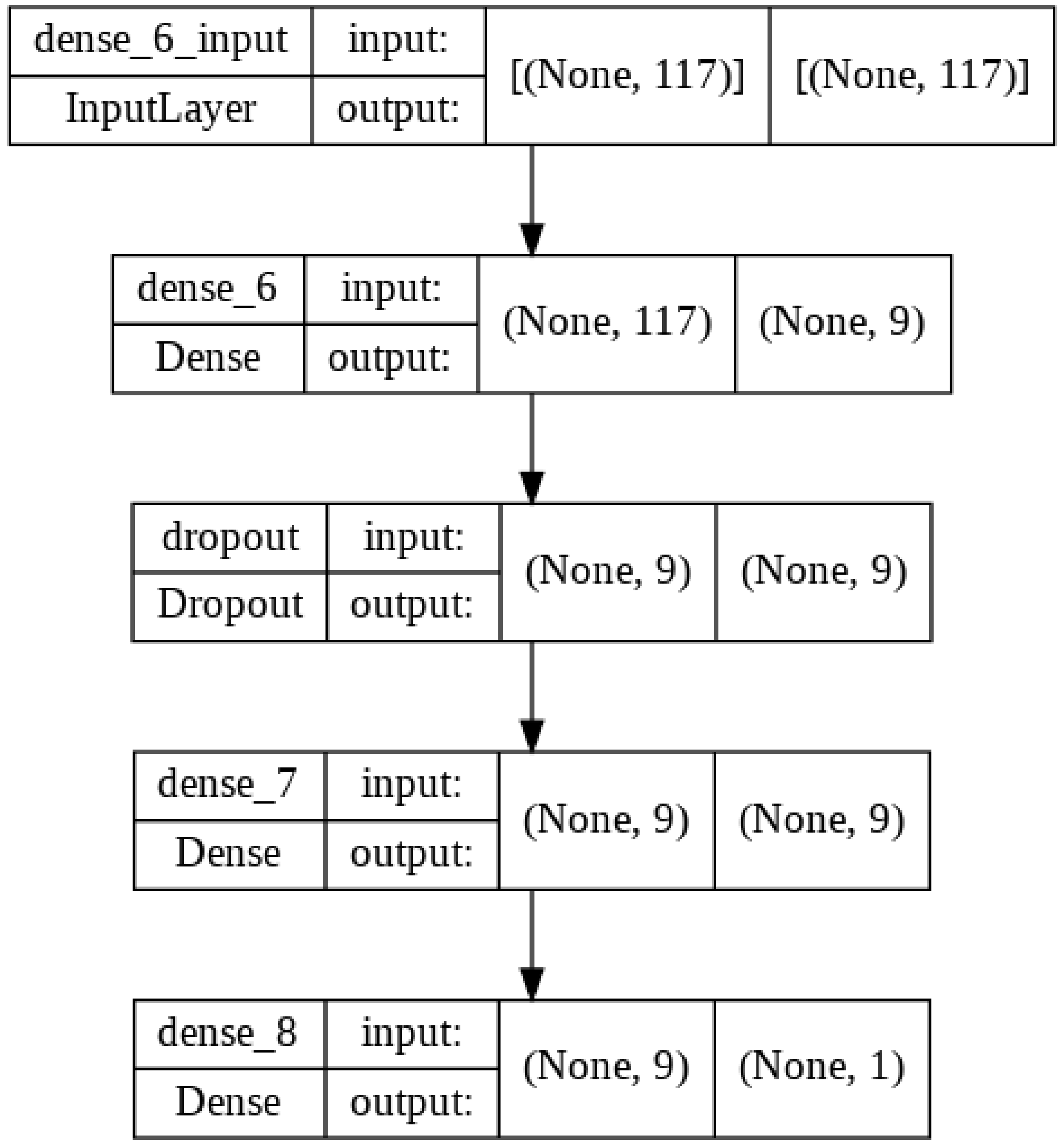


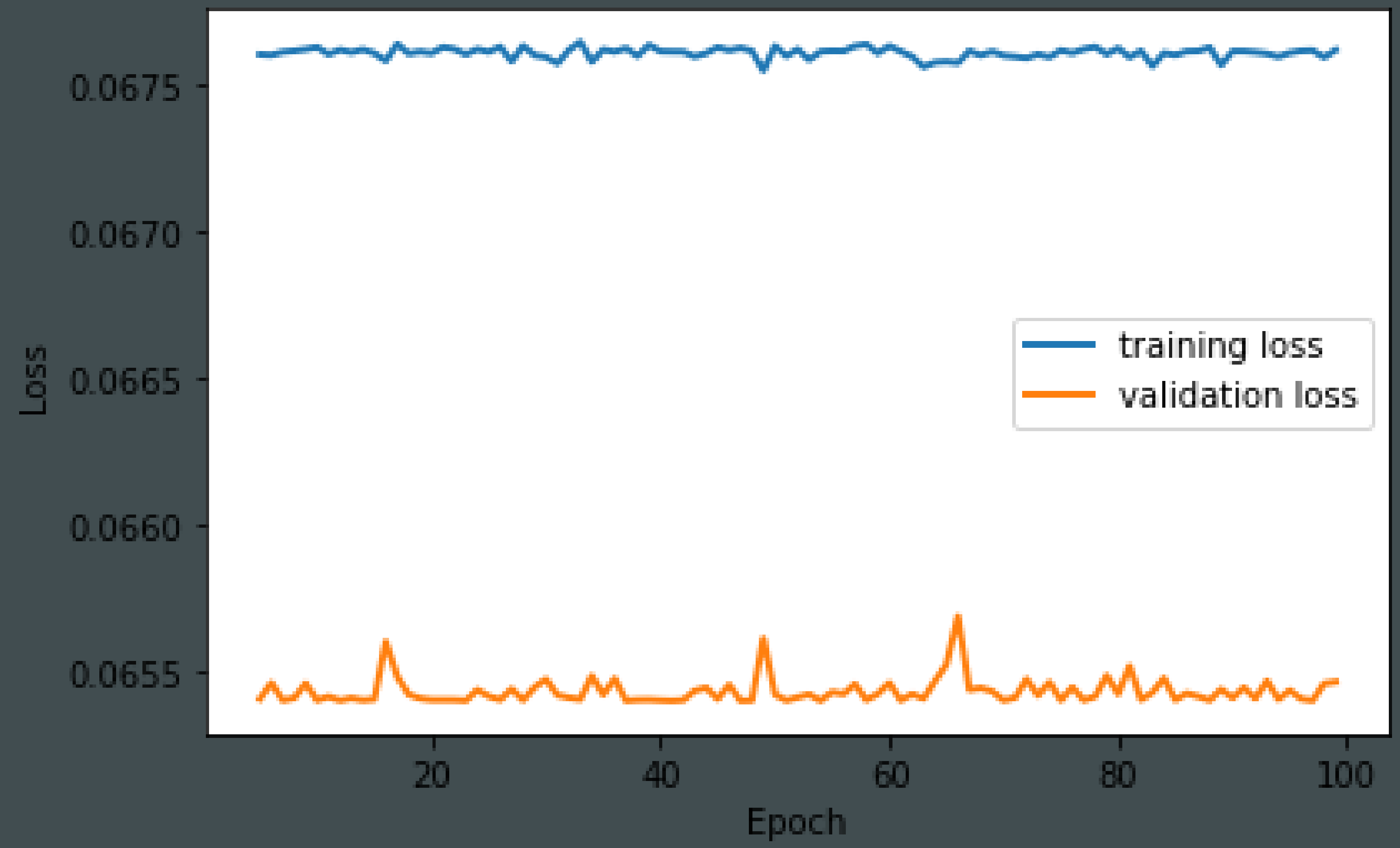
- Tree Based Algo
- Tuned Model
- Normalised Data set

- Potential Solution Reuse
- Automated pipeline for model training and evaluation

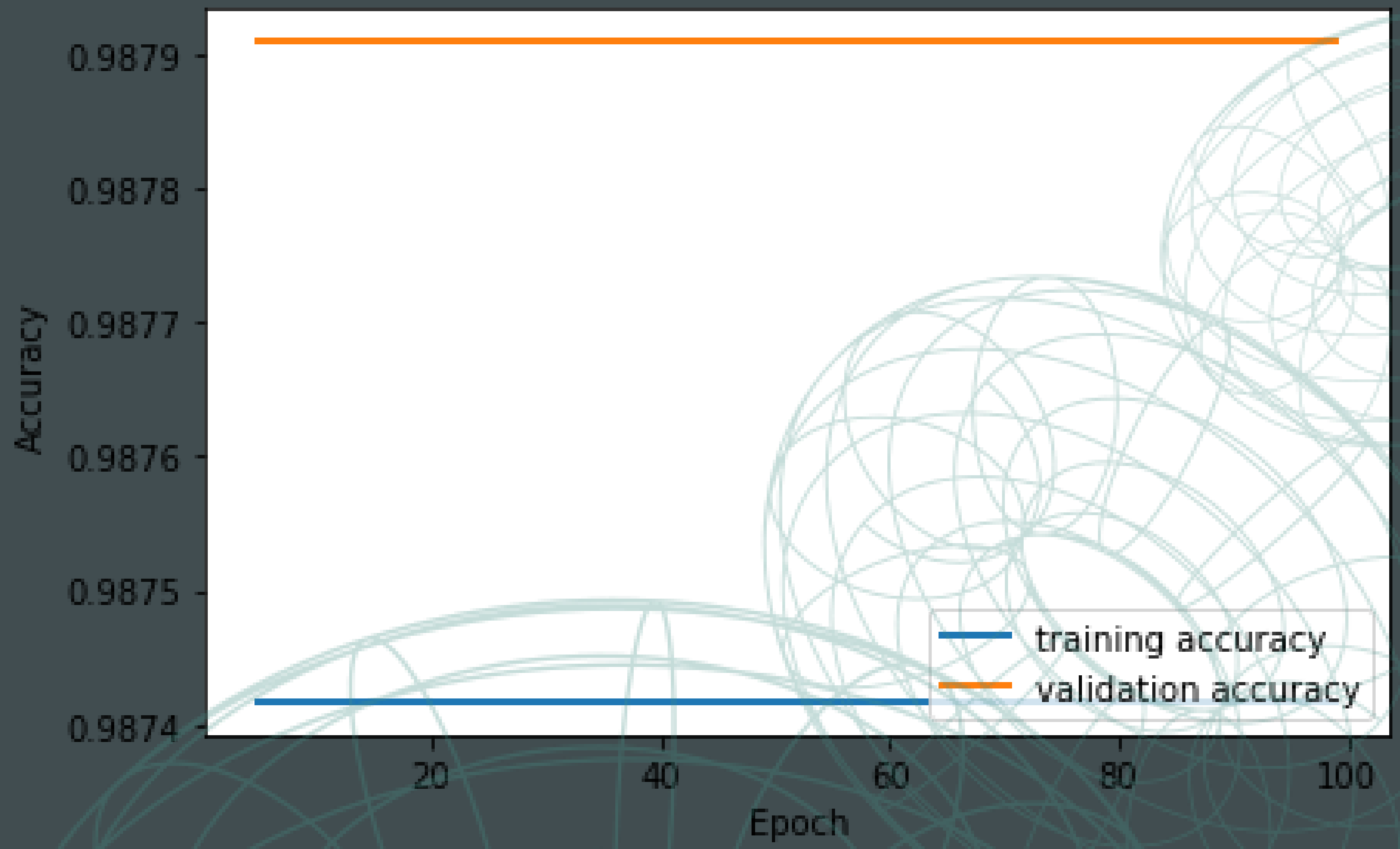
- **RLF can be predicted with good reliability using weather coditions, RL KPI metrics and terrain characteristics**
- **Frequent re-training with new RL failure scenarios will improve model performance**

Deep learning model





0.0675
0.0670
0.0665
0.0660
0.0655
Loss
20
40
60
80
100
Epoch
training loss
validation loss



0.9879
0.9878
0.9877
0.9876
0.9875
0.9874
Accuracy
20
40
60
80
100
Epoch
training accuracy
validation accuracy

Machine learning vs Deep learning

	Machine learning	Deep learning
Run time	0,64 sec	0,75sec
Accuracy	99,7%	98,7%

Conclusion

RLF prediction will augment self healing capability of networks by enabling them to take further actions to avoid service interruption / degradation.


Thank you!