LGBM Classifier for Fusion Energy Multi-Machine Disruption Prediction

ITU AI-5G Challenge

Isaac Oluwafemi Ogunniyi

Abstract

This work addresses the challenge of the global energy crisis by improving the prospects of one of the world's efficient and sustainable energy sources; nuclear fusion, by tackling the unwanted occurrence of disruptions in the nuclear fusion process across different types of nuclear fusion machines (tokamaks).
The approach of this work is to train a Light Gradient Boosting Machine Classifier on features engineered from the line integral density (center chord) diagnostic channel of the tokamak.
This approach achieves an impressive f1 score of 91.3 in predicting the occurrence of disruption in a tokamak type it was not trained on.

1. Introduction

Nuclear fusion involves the merging of two light nuclei, a process that releases a substantial amount of energy. This fusion occurs within a plasma, a state of matter consisting of ions and free electrons, and represents a clean, safe, and accessible energy source. One method of achieving nuclear fusion involves the confinement of the two light nuclei by the use of a torus-shaped magnetic field within devices known as tokamaks.

The problem however impeding progress in this technology are disruptions: instabilities that arise within the tokamak plasma, leading to energy losses and termination of the fusion process together with other devastating effects. It will therefore be hugely impactful to be able to predict these disruptions before they occur.

The objective of this work is to use machine learning to develop a model that can learn from diagnostic signals in a tokamak and predict when a disruption is likely to occur so that they can be averted. This work is also very important because patterns learnt from disruptions in currently existing tokamaks can be used to predict disruptions in even future tokamaks.

To achieve this objective this work makes use of data of three different types of tokamaks: HL-2A, J-TEXT and Alcator CMod. The J-TEXT and HL-2A tokamaks are used as the current devices and Alcator C-Mod as the future device.

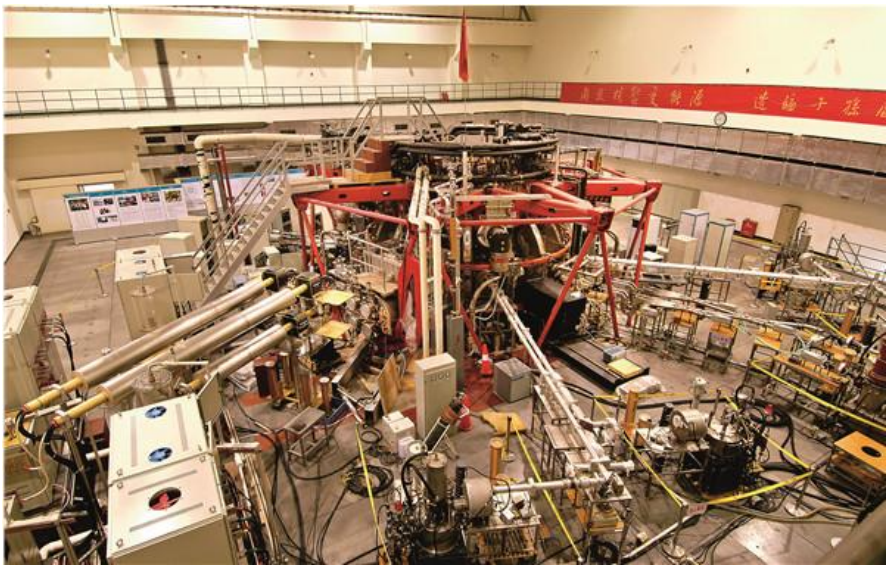

Figure 1. HL-2A tokamak of SWIP Chengdu, China [1]

2. Dataset

The data used in this work was accessed from Zindi and ITU [2]. It is made up of 975 shot files from the HL-2A tokamak, 2136 shot files from the J-TEXT tokamak and 433 shot files from the Alcator C-Mod tokamak such that 20 of the 433 Alcator C-Mod shot files are used for validation of the performance and 413 for the final test.

For each of these sets of shot files stored in the hdf5 format, the contents were diagnostic signals and the metadata on these diagnostic signals.
The table below shows a sample of the 94 different diagnostic signals recorded for each tokamak. Signals measuring similar characteristics across different tokamaks share the same row.

Table 1. The various kinds of diagnostic signals present in a shot file

Diagnostics	Type	Sampling rate(kHz)	J-TEXT MDSplus Tag	C-Mod MDSplus Tag	HL-2A MDSplus Tag
plasma current	Raw	1	ip	\ip	CCO-LFB:LFEX-IP
toroidal magnetic field	Raw	1	bt	\btor	CCO-LFB:LFBBT
horizontal displacement	Raw	1	dx	\efit_aeqdsk:rmagx	CCO-LFB:LFDH
vertical displacement	Raw	1	dy	\efit_aeqdsk:zmagx	CCO-LFB:LFDV
line integral density(center chord)	Raw	1	polaris_den_v09	.tci.results:nl_04	CCO-DF:DENSITY1
line integral density(high field side)	Raw	1	polaris_den_v01	N/A	N/A
line integral density(low field side)	Raw	1	polaris_den_v17	N/A	N/A
C3 radiation	Raw	1	vs_c3_aa018	\twopi_diode	N/A
loop voltage	Raw	1	vl	\efit_aeqdsk:vloopt	DS-EMD-ROG:VL-FILTER
AXUV_01	Raw	1	AXUV_CA_02	\AXA.chord_1 & \AXJ.chord_1	DS-BM-AB:BOLD01

3. Feature Engineering

The dataset contains 94 different sets of signals which had the potential of being used as features to train the models, the signal of line integral density (center chord) was however chosen based on the following reasons:

1. Availability of observations of the signal across the three different tokamak types. Not all signals had values recorded across the different tokamak types. Out of the 94, only 61 had all three sets of observations present.
   Some of these are:
   - Plasma current
   - Toroidal magnetic field
   - Horizontal displacement
   - Vertical displacement
   - Line integral density(center chord)
   - Loop voltage
   - Soft X-ray 01
   - Soft X-ray 02
   - Poloidal Mirnov probe
   - Rotating mode proxy

2. For some signals with observations across all three tokamaks, some instances of the shot files in the test document do not have that signal present.

3. Lastly, only a few signals had information that models could learn and accurately generalize across different tokamaks.The table below shows some diagnostic channels which had observations present across various tokamaks and the extent to which information learnt from them aided a model predict disruptions in tokamaks it was not trained on.

   Table 2. Generalization performance of a baseline logistic regression model on 4 different signal data

Signal	Training tokamak score	Unseen tokamak score
Line integral (center chord)	0.8805	0.6250
Loop voltage	0.8793	0.6

Soft X-ray 05	0.8805	0.5385
Plasma current	0.8711	0.5185

The following are the additional features that were engineered:

1. Statistical measures of the observations in the line integral density(center chord) signal:
    a. Population size
    b. Mean
    c. Range (peak-to-peak)
    d. Standard deviation
    e. Median
    f. Maximum
    g. Minimum
    h. Interquartile range
    i. variance
2. Same statistical measures as above but of a one-step differenced version of the observations in the line integral density (center chord) signal.

3. A boolean value representing the validity of the diagnostic signal of that particular shot file
4. A one-hot encoding of the type of tokamak

4. Proposed Model

4.1 Model Structure

The proposed model is structured as a two-step scikit-learn pipeline:

1. A StandardScaler object for scaling the data and making it more suitable for machine learning algorithms.

2. An LGBMClassifier instance to fit and learn the complex relationships between the engineered features and the occurrence of disruptions.

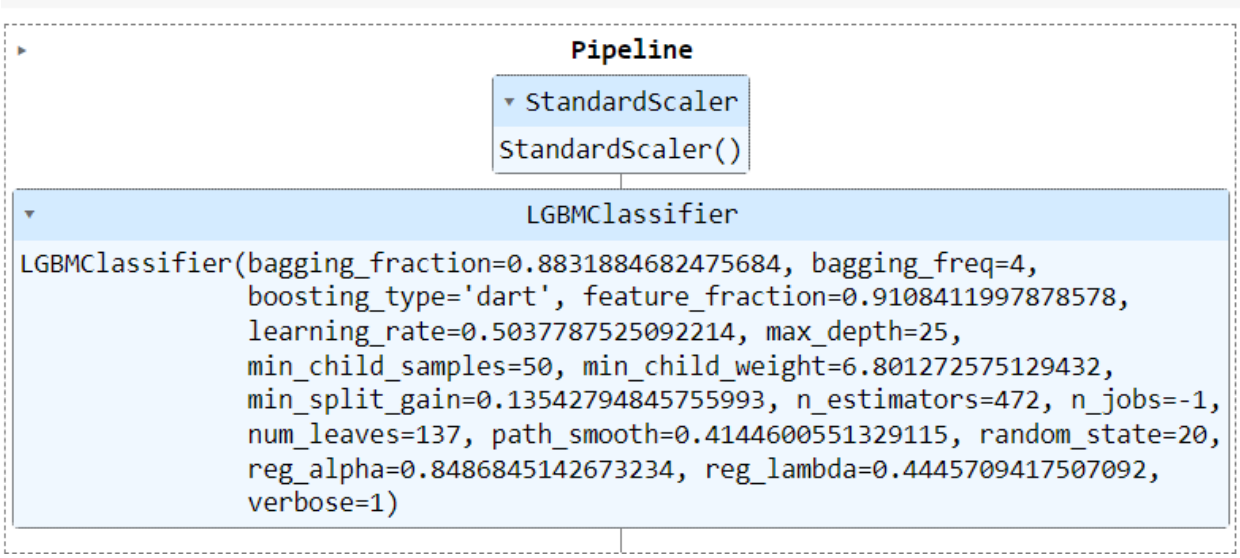

Figure 2. Structure of the Proposed Model

4.2 Training the Model

The choice of LGBMClassifier was made as a result of searching across a number of models. To streamline the model search process, a function was created that allowed multiple models to be trained on the same data. The code snippet below shows the function.

```
# Defining function that will run the hyperparameter tuning
def runmodel(model,tuning_params,scorer=make_scorer(f1_score),n_iter=5):
  pipe = Pipeline(steps=[
        ('sc',StandardScaler()),
          ('classifier',model)
        ])
  r_search = RandomizedSearchCV(pipe,tuning_params,n_jobs=-1,verbose=-1,
                                scoring=scorer,cv=10,n_iter=n_iter,
                                random_state=2,error_score='raise')
  r_search.fit(train.drop('Is_disrupt',axis=1),train['Is_disrupt'])
  return r_search
```

Code snippet 1. Custom function to run model fitting and hyperparameter tuning

The search process involved fitting each model to the data and tuning the hyperparameters by use of RandomizedSearchCV. In total, 11 models were trained and evaluated on the data.

4.3 Evaluating the Model

The table below shows the result of evaluating the several models on the train and validation sets of data.

Table 3. Evaluation results (RandomizedSearchCV) of models (F1_scores)

	Model	Test Scores	Train Scores
0	LGBMClassifier	1.000000	0.933257
1	GradientBoostingClassifier	1.000000	0.931032
2	XGBoostClassifier	1.000000	0.929035
3	RandomForestClassifier	1.000000	0.927909
4	DecisionTreeClassifier	1.000000	0.910767
5	BaggingClassifier	1.000000	0.904632
6	VotingClassifier	0.769231	0.898374
7	LogisticRegression	0.625000	0.880523
8	AdaBoost	0.600000	0.936626
9	SGDClassifier	0.588235	0.901149
10	SVC	0.588235	0.898186
11	KNeighbors Classifier	0.000000	0.739614
12	ExtraTreesClassifier	0.000000	0.000000

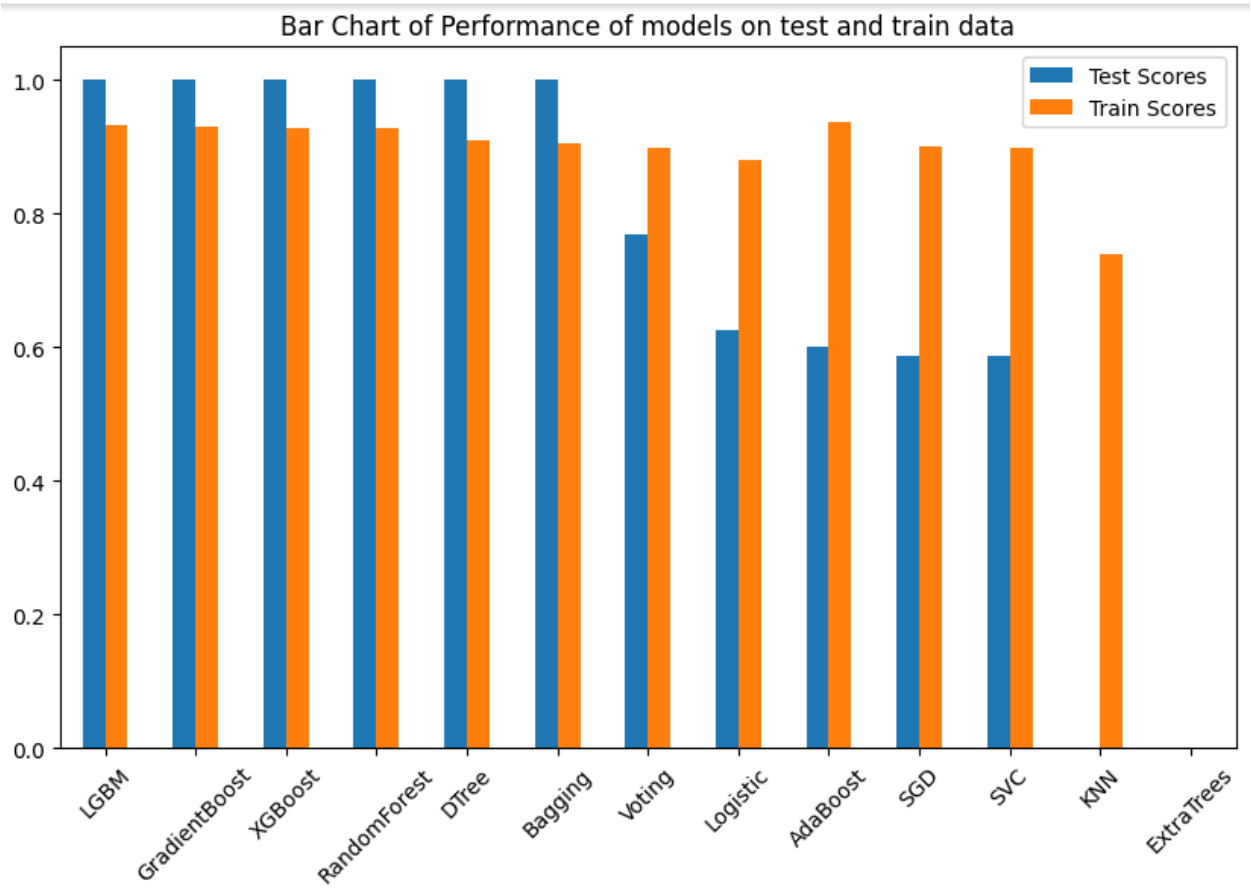

Figure 3. Bar chart showing performance (RandomizedSearchCV) of 13 models on test and train data

Based on the evaluation of the various models, as seen above, LGBMClassifier was selected due to its prediction accuracy on the train and validation data. After the evaluation on the final set of test data, the model predicted with an impressive accuracy score of 0.991071428.
Below is a figure with a horizontal bar plot showing the 10 features that the model depended on the most for predicting the occurrence of disruptions:

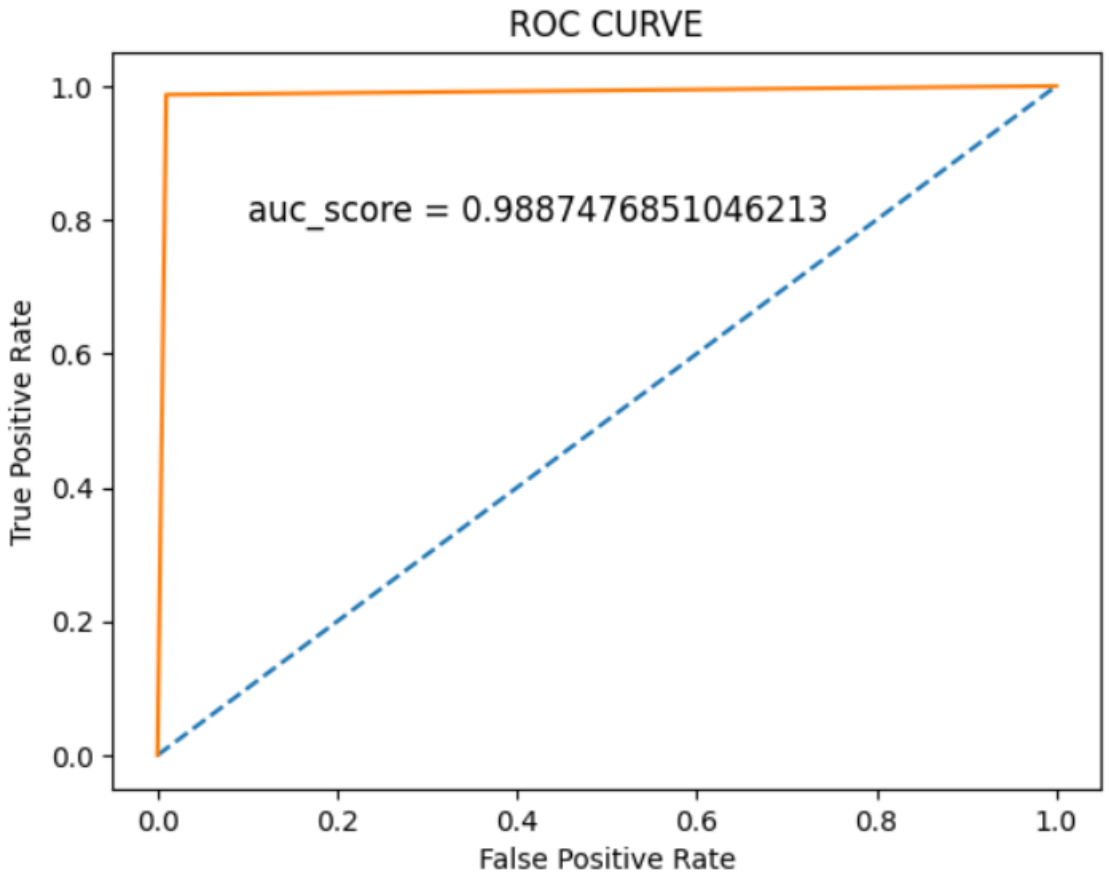

Figure 4. ROC curve with AUC score of LGBMClassifier Predictions on train data

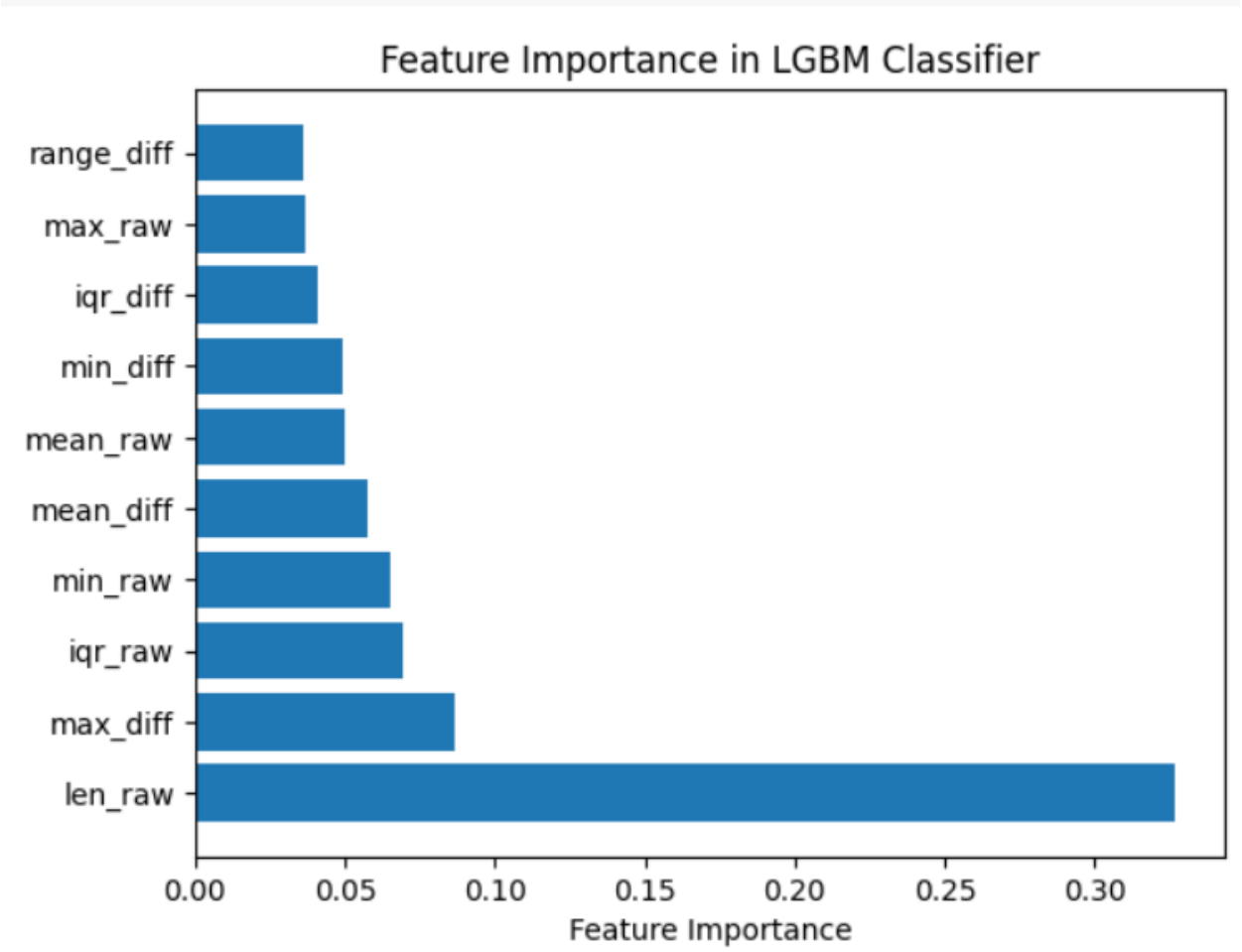

Figure 4. Top 10 important features for predicting Disruptions

5. Conclusion and Future Work

This solution demonstrates an effective approach to predicting disruptions in a tokamak using machine learning. By utilizing data from the line integral density diagnostic signal, feature engineering, and an LGBMClassifier model, accurate predictions can be made about the occurrence or otherwise of disruption within a tokamak.

The main packages used in the Python environment for this project are:

- pandas (1.5.3) [3]
- numpy (1.23.5) [4]
- scikit-learn (1.2.2) [5]
- scipy (1.11.3) [6]
- lightgbm (4.1.0) [7]
- h5py (3.9.0) [8]

- google (2.0.3) [9]

The notebook can be executed on Google Colab with a CPU runtime, and the entire process takes less than 6 minutes.

Future work should consider making use of data from multiple signals per shot file either averaged and treated as tabular data or combined a matrix and a neural network trained on it.

References

1. HL-2A
2. Multi-Machine Disruption Prediction Challenge for Fusion Energy - Zindi
3. What's new in 1.5.3 (January 18, 2023) — pandas 2.1.1 documentation
4. NumPy 1.23.5 Release Notes
5. Version 1.2.2 — scikit-learn 1.3.1 documentation
6. Release Notes — SciPy v1.11.3 Manual
7. lightgbm · PyPI
8. h5py · PyPI
9. google · PyPI