Project Phase Report - 1

INTP22-ML-5: Power Line Fault Detection

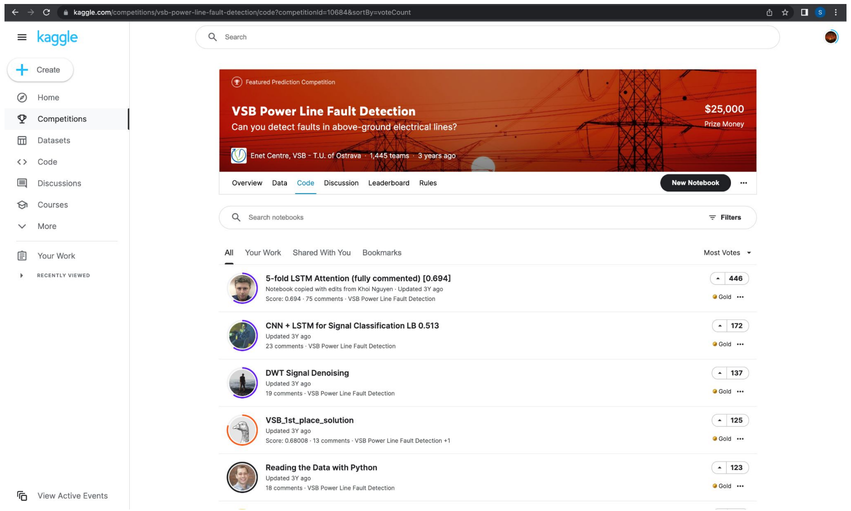
Sshubam Verma

PROJECT OBJECTIVES FOR THE PHASE:

- Perform a deep dive into the data through Exploratory Data Analysis
- Plot various Signal Data samples using Seaborn
- Learn optimal Preprocessing techniques to Apply on the Data
- Analyse existing solutions and techniques to learn and explore existing techniques, to develop a better solution
- Go through Research papers to get a better understanding of the approach to the problem

PHASE - 1 PROGRESS

I researched about the problem statement and went through a bunch of resources to refer, first kaggle where I found various notebooks and modelling approaches for this problem statement. I went through some notebooks to get a better understanding of what modelling techniques are being used already to develop a better model. I also downloaded the dataset and now working on exploratory data analysis and getting familiar with the data and manipulating parquet data in python through the given resources. I learned about how I should transform the data iteratively and techniques to build a Neural network with large amount of data to avoid memory issues, because I ran into one.



Featured Prediction Competition

VSB Power Line Fault Detection

Can you detect faults in above-ground electrical lines?

$25,000 Prize Money

Enet Centre, VSB - T.U. of Ostrava · 1,445 teams · 3 years ago

5-fold LSTM Attention (fully commented) [0.694] — 446 — Gold
Notebook copied with edits from Khoi Nguyen · Updated 3Y ago
Score: 0.694 · 75 comments · VSB Power Line Fault Detection

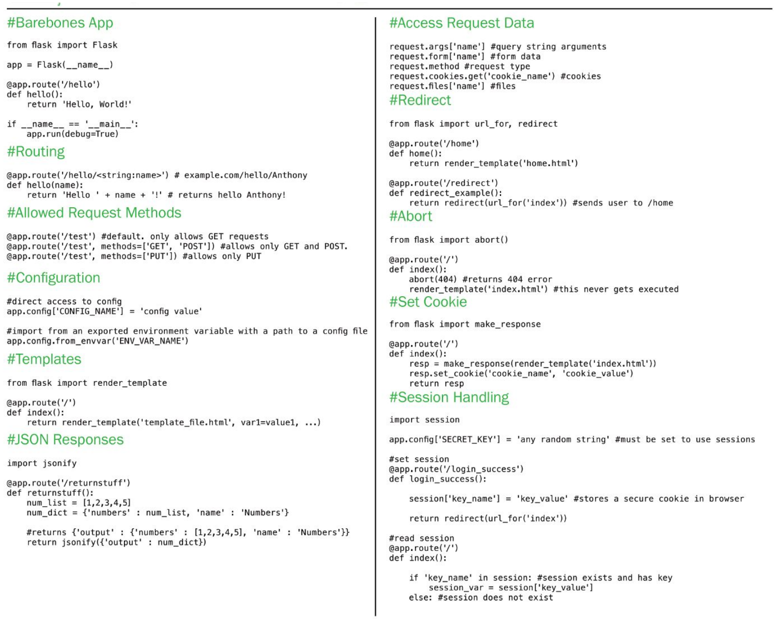
CNN + LSTM for Signal Classification LB 0.513 — 172 — Gold
Updated 3Y ago
23 comments · VSB Power Line Fault Detection

DWT Signal Denoising — 137 — Gold
Updated 3Y ago
19 comments · VSB Power Line Fault Detection

VSB_1st_place_solution — 125 — Gold
Updated 3Y ago
Score: 0.68008 · 13 comments · VSB Power Line Fault Detection +1

Reading the Data with Python — 123 — Gold
Updated 3Y ago
18 comments · VSB Power Line Fault Detection

PHASE - 1 PROGRESS



#Barebones App

```
from flask import Flask

app = Flask(__name__)

@app.route('/hello')
def hello():
    return 'Hello, World!'

if __name__ == '__main__':
    app.run(debug=True)
```

#Routing

```
@app.route('/hello/<string:name>') # example.com/hello/Anthony
def hello(name):
    return 'Hello ' + name + '!' # returns hello Anthony!
```

#Allowed Request Methods

```
@app.route('/test') #default. only allows GET requests
@app.route('/test', methods=['GET', 'POST']) #allows only GET and POST.
@app.route('/test', methods=['PUT']) #allows only PUT
```

#Configuration

```
#direct access to config
app.config['CONFIG_NAME'] = 'config value'

#import from an exported environment variable with a path to a config file
app.config.from_envvar('ENV_VAR_NAME')
```

#Templates

```
from flask import render_template

@app.route('/')
def index():
    return render_template('template_file.html', var1=value1, ...)
```

#JSON Responses

```
import jsonify

@app.route('/returnstuff')
def returnstuff():
    num_list = [1,2,3,4,5]
    num_dict = {'numbers' : num_list, 'name' : 'Numbers'}

    #returns {'output' : {'numbers' : [1,2,3,4,5], 'name' : 'Numbers'}}
    return jsonify({'output' : num_dict})
```

#Access Request Data

```
request.args['name'] #query string arguments
request.form['name'] #form data
request.method #request type
request.cookies.get('cookie_name') #cookies
request.files['name'] #files
```

#Redirect

```
from flask import url_for, redirect

@app.route('/home')
def home():
    return render_template('home.html')

@app.route('/redirect')
def redirect_example():
    return redirect(url_for('index')) #sends user to /home
```

#Abort

```
from flask import abort()

@app.route('/')
def index():
    abort(404) #returns 404 error
    render_template('index.html') #this never gets executed
```

#Set Cookie

```
from flask import make_response

@app.route('/')
def index():
    resp = make_response(render_template('index.html'))
    resp.set_cookie('cookie_name', 'cookie_value')
    return resp
```

#Session Handling

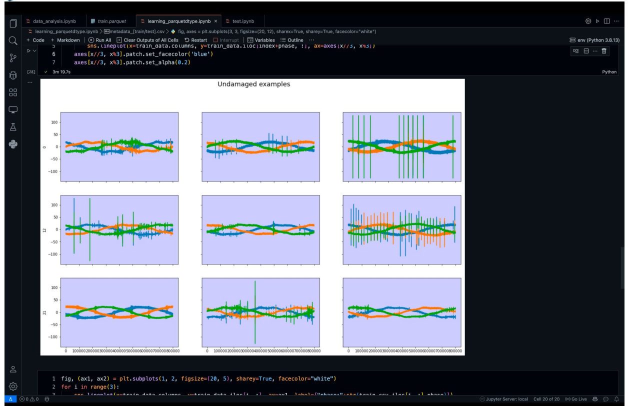
```
import session

app.config['SECRET_KEY'] = 'any random string' #must be set to use sessions

#set session
@app.route('/login_success')
def login_success():

    session['key_name'] = 'key_value' #stores a secure cookie in browser

    return redirect(url_for('index'))

#read session
@app.route('/')
def index():

    if 'key_name' in session: #session exists and has key
        session_var = session['key_value']
    else: #session does not exist
```

Since I have to deploy my web-app in Flask including the model, I started learning it from various videos and the official Flask documentation, I am currently learning about how to implement endpoints, requests and how to feed the data into the deployed model for making prediction and how to use the returned output to display. I have an Idea to implement a form to enter signal information in the Flask App, which will be integrated with the trained model for inference. The Flask app can be deployed to Heroku cloud service.

PHASE - 1 PROGRESS

I found two research papers and went through them (1: https://github.com/randxie/Kaggle-VSB-Baseline/blob/master/papers/Thesis%20Partial%20Discharge.pdf) (2: https://github.com/randxie/Kaggle-VSB-Baseline/blob/master/papers/On-line%20Signal%20Analysis%20of%20Partial%20Discharges%20in%20Medium%20Voltage%20Power%20Cables.pdf),
in which I learned about the need for solving this problem, about Partial discharge, how it related to causing a Power Line Fault and how it is formed and how to use machine learning to solve this problem using optimal preprocessing for signal data. I am currently learning about advanced preprocessing for this type of signal data and how to deal with noise in signal data.I also learned about Parquet data and how is it efficient in case of big data problems and how it is optimized for working with bulk and complex data as Parquet can only read the columns needed and therefore reduces the IO, I also learned how to manipulate Parquet data for loading.

PHASE - 1 PROGRESS



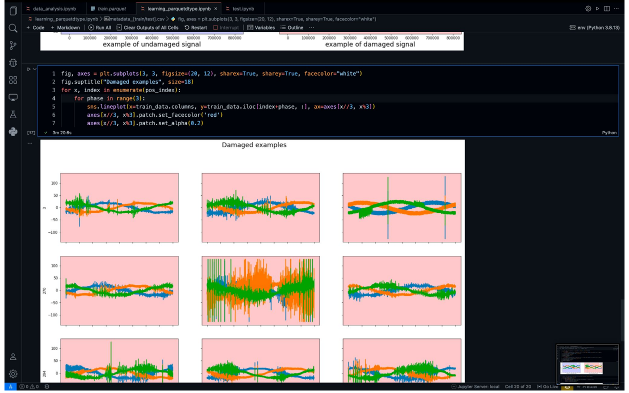
data_analysis.ipynb
train.parquet
learning_parquetdtype.ipynb
test.ipynb
learning_parquetdtype.ipynb > metadata_[train/test].csv > fig, axes = plt.subplots(3, 3, figsize=(20, 12), sharex=True, sharey=True, facecolor="white")
Code
Markdown
Run All
Clear Outputs of All Cells
Restart
Interrupt
Variables
Outline
env (Python 3.8.13)
Python
Jupyter Server: local
Cell 20 of 20
Go Live

```
        sns.lineplot(x=train_data.columns, y=train_data.iloc[index+phase, :], ax=axes[x//3, x%3])
    axes[x//3, x%3].patch.set_facecolor('blue')
    axes[x//3, x%3].patch.set_alpha(0.2)
```

[28] 3m 19.7s

Undamaged examples

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(20, 5), sharey=True, facecolor="white")
for i in range(3):
```



data_analysis.ipynb
train.parquet
learning_parquetdtype.ipynb
test.ipynb
learning_parquetdtype.ipynb > metadata_[train/test].csv > fig, axes = plt.subplots(3, 3, figsize=(20, 12), sharex=True, sharey=True, facecolor="white")
Code
Markdown
Run All
Clear Outputs of All Cells
Restart
Interrupt
Variables
Outline
env (Python 3.8.13)
example of undamaged signal
example of damaged signal
Python

```
fig, axes = plt.subplots(3, 3, figsize=(20, 12), sharex=True, sharey=True, facecolor="white")
fig.suptitle("Damaged examples", size=18)
for x, index in enumerate(pos_index):
    for phase in range(3):
        sns.lineplot(x=train_data.columns, y=train_data.iloc[index+phase, :], ax=axes[x//3, x%3])
        axes[x//3, x%3].patch.set_facecolor('red')
        axes[x//3, x%3].patch.set_alpha(0.2)
```

[37] 3m 20.6s

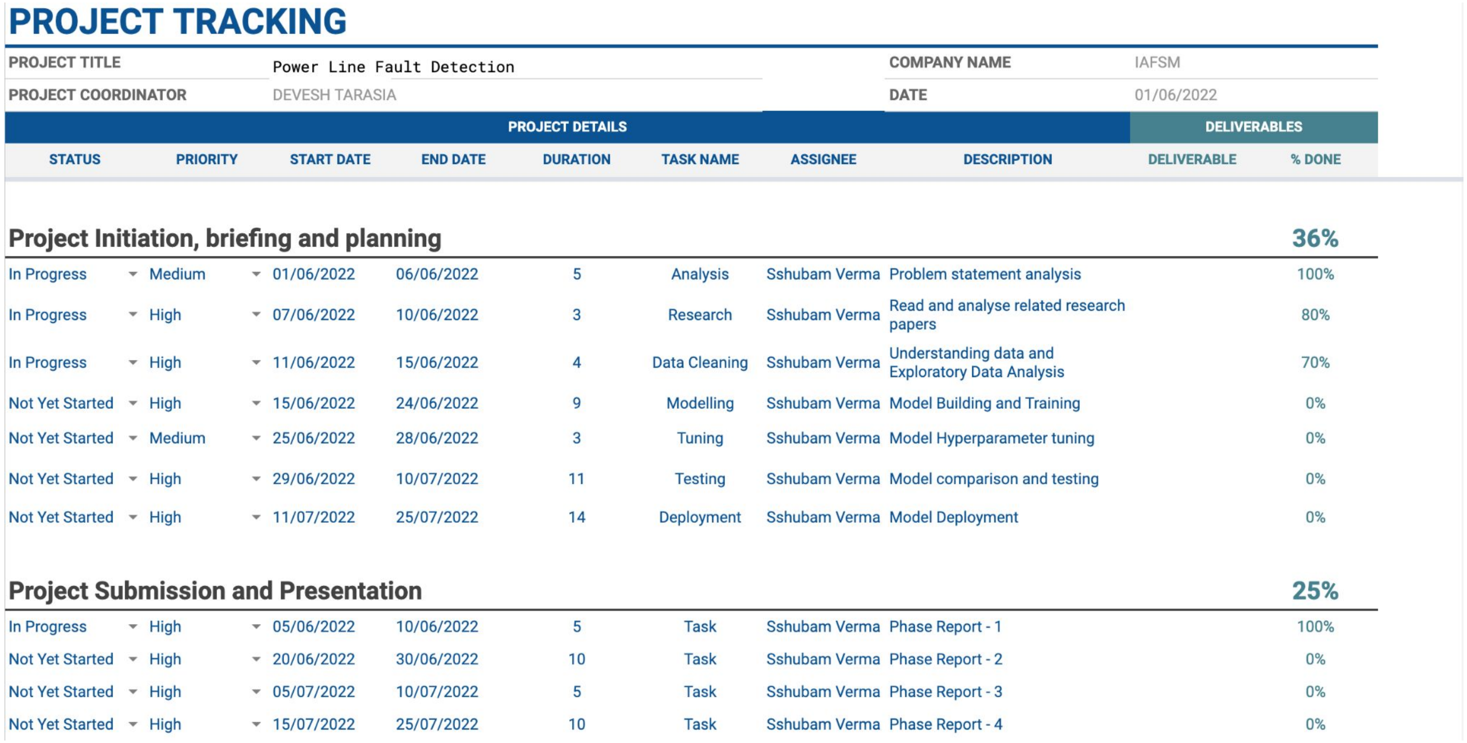
Damaged examples

I performed Exploratory Data Analysis on the Signal Data using Seaborn to get familiar with the data I am working on, to implement the best model to this data. I learned how I have to reduce the noise in the Data before feeding it to the model or extract significant features in the signals. I plotted some of the data from our dataset to get a better understanding of the values of voltage if a power line has a fault or not, I also observe some noise which needs to be eliminated or reduced in some way, on which I am working to explore and learn methods to preprocess this signal data.

[Notebook Link]

Gantt Chart

Gantt Chart LINK



PROJECT TRACKING

PROJECT TITLE	Power Line Fault Detection	COMPANY NAME	IAFSM
PROJECT COORDINATOR	DEVESH TARASIA	DATE	01/06/2022

PROJECT DETAILS								DELIVERABLES	
STATUS	PRIORITY	START DATE	END DATE	DURATION	TASK NAME	ASSIGNEE	DESCRIPTION	DELIVERABLE	% DONE
Project Initiation, briefing and planning									**36%**
In Progress	Medium	01/06/2022	06/06/2022	5	Analysis	Sshubam Verma	Problem statement analysis		100%
In Progress	High	07/06/2022	10/06/2022	3	Research	Sshubam Verma	Read and analyse related research papers		80%
In Progress	High	11/06/2022	15/06/2022	4	Data Cleaning	Sshubam Verma	Understanding data and Exploratory Data Analysis		70%
Not Yet Started	High	15/06/2022	24/06/2022	9	Modelling	Sshubam Verma	Model Building and Training		0%
Not Yet Started	Medium	25/06/2022	28/06/2022	3	Tuning	Sshubam Verma	Model Hyperparameter tuning		0%
Not Yet Started	High	29/06/2022	10/07/2022	11	Testing	Sshubam Verma	Model comparison and testing		0%
Not Yet Started	High	11/07/2022	25/07/2022	14	Deployment	Sshubam Verma	Model Deployment		0%
Project Submission and Presentation									**25%**
In Progress	High	05/06/2022	10/06/2022	5	Task	Sshubam Verma	Phase Report - 1		100%
Not Yet Started	High	20/06/2022	30/06/2022	10	Task	Sshubam Verma	Phase Report - 2		0%
Not Yet Started	High	05/07/2022	10/07/2022	5	Task	Sshubam Verma	Phase Report - 3		0%
Not Yet Started	High	15/07/2022	25/07/2022	10	Task	Sshubam Verma	Phase Report - 4		0%