

Lithofacies Classification using Supervised Machine Learning

by: Stefanus Yudi Irwan

Final Project
Course: Introduction to Machine Learning
& Machine Learning Process
Sekolah Data Pacmann

1. Problem Definition & Goals
2. Project Timeline
3. Data Preparation
4. Data Preprocessing
5. Feature Engineering
6. Modeling
7. Model Evaluation
8. Front End and Back End Services
9. Pytest
10. Deployment

Problem Definition & Goals



Business Problems

- **Oil and Gas companies** need to translate **well measurement data** into lithofacies layer to **better understand** the condition of the **reservoir** being drilled.
- **Manually interpreting** well measurement data that are exponentially growing in volume by reservoir **geologists or geophysicists** must be **subjective** to some extent, leading to **increased uncertainties**.
- Facies definition is sometimes very **time-consuming** activity and **expensive**.



Business Solution

- **Classification of Lithofacies** can be achieved by using **supervised machine learning technique**. This supervised technique used **lithofacies labeled data** to understand the patterns and then **label other data lithofacies** based on trained lithofacies patterns
- In this **research and deployment** we will construct **supervised machine learning** model to **classify lithofacies** using **well-measurement data** to reduce cost and tackle the uncertainty of manual interpretation

- The **goal** of this project is to find the **best-supervised machine learning algorithm** for lithofacies classification, and then **deploy the pre-application** to the server to predict the lithofacies from the well-measurement data

Machine Learning Metrics

- 1. Accuracy**
0.5 – 0.6
How well does the model predicts the true positive and true negative labels from the data input
- 2. Adjacent Accuracy**
0.6-0.8
How well does the model predicts the adjacent facies of the labels
- 3. CV Score**
0.5-0.6
How is the model performance through training and validation data
- 4. ROC-AUC Value**
0.8-0.9
How well the model can separate the True Positive and False Positive

Business Metrics

- 1. Cost**
Cost that was spent to interpret the well measurement data
- 2. Work Execution Time**
Time spent to interpret the well measurement data

Project Timeline

Project Timeline

Month	Week	Project Topic	Data Preparation	Data Preprocessing	Feature Engineering	Modeling	API Services	Front End Services	Docker Services	Deployment Services	Submission
Oct-22	1										
	2										
	3										
	4										
Nov-22	1										
	2										
	3										
	4										
Dec-22	1										

- **Project Timeline** : October Week 1 – December Week 1
- **Project Steps** : Topic → Data Preparation → Data Processing → Feature Engineering → Modeling → API Services → Front End Services → Docker Services → Deployment → Submission and Reporting

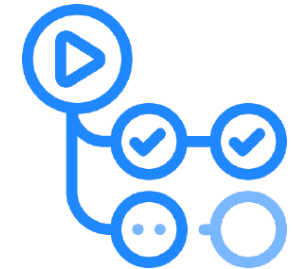
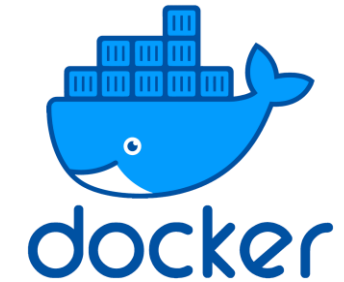
Developing ML



Front End & Back End



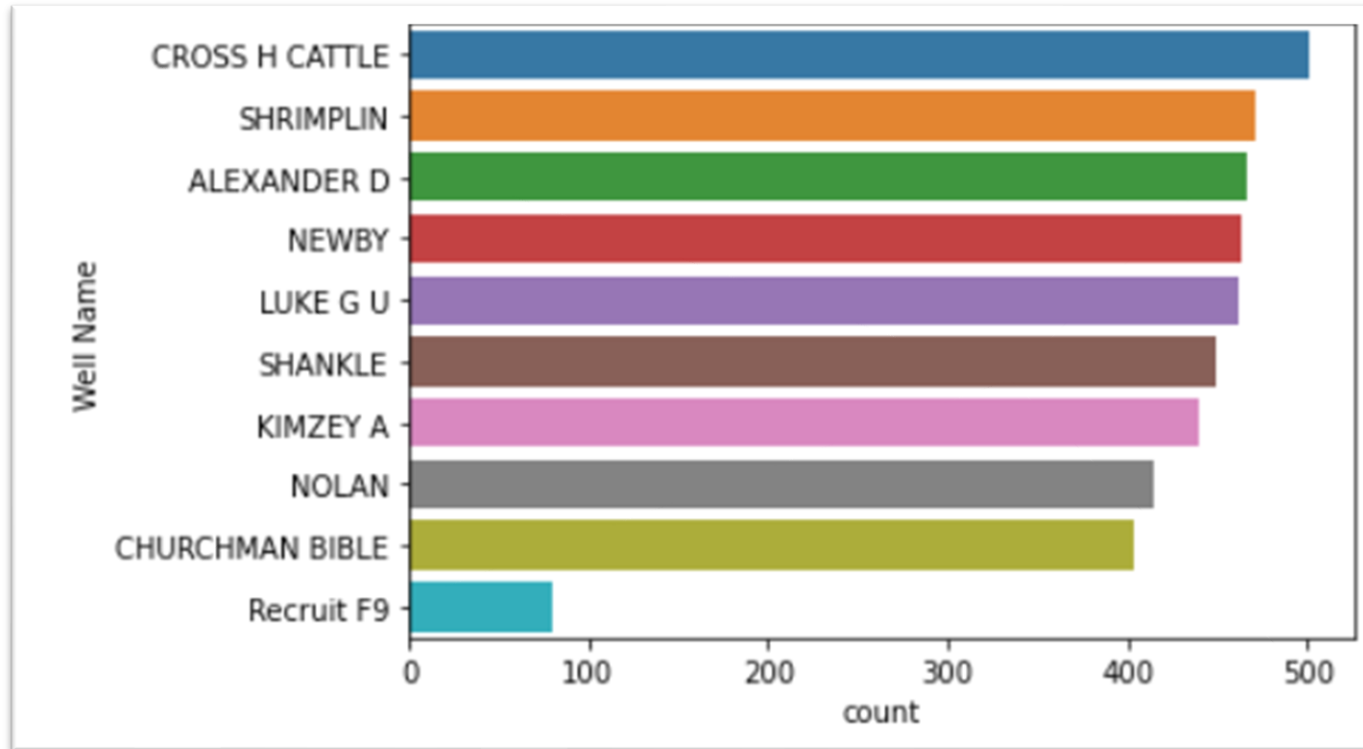
Deployment



Data Preparation

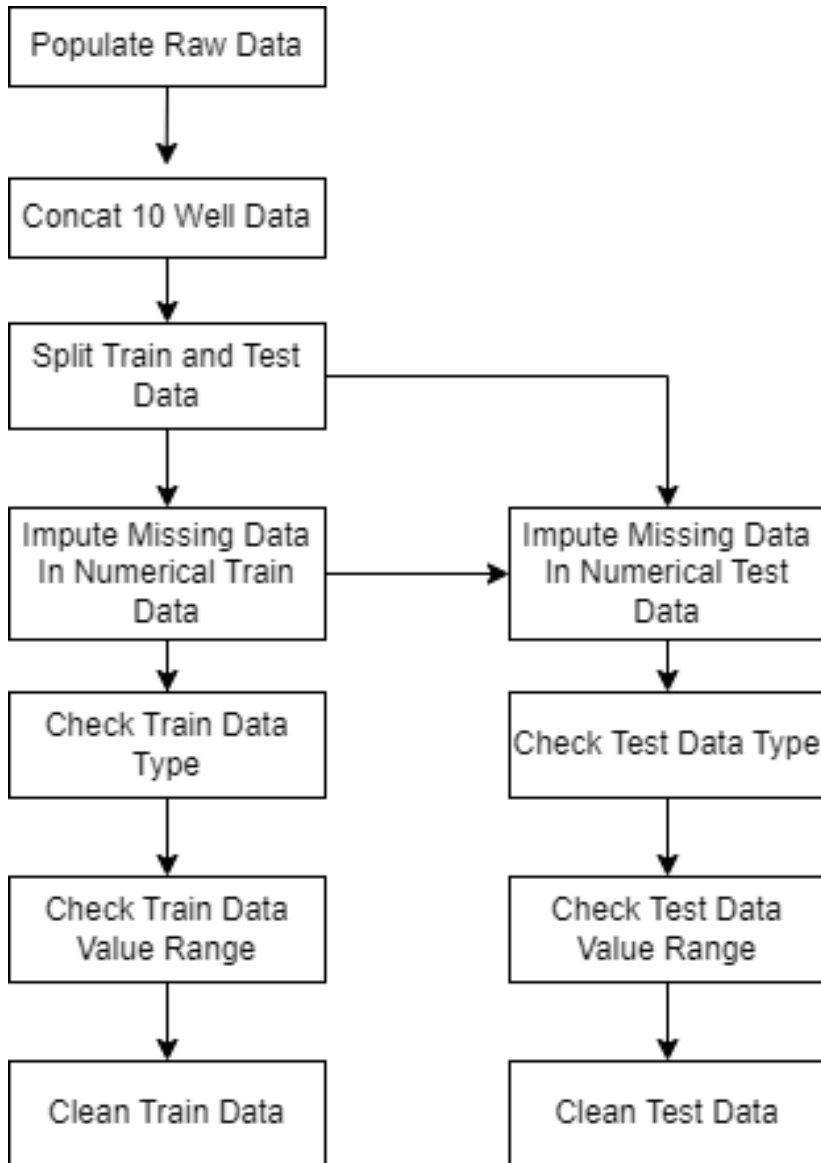
- Dataset are from [Machine Learning Competition in 2016](#)
- Dataset comprises **11 columns** and **4149 rows**
- There are **3 categorical data**: Facies, Formation, and Well Name
- There are **7 numerical data**: Depth, GR, ILD_log10, Delta-PHI, PHIND, PE, NM_M, RELPOS
- Numerical data consist of **5 Wireline Measurement** and **2 Geological Variable**

	Facies	Formation	Well Name	Depth	GR	ILD_log10	DeltaPHI	PHIND	PE	NM_M	RELPOS
0	CSiS	A1 SH	NOLAN	2853.5	106.813	0.533	9.339	15.222	3.500	1	1.000
1	FSiS	A1 SH	NOLAN	2854.0	100.938	0.542	8.857	15.313	3.416	1	0.977
2	FSiS	A1 SH	NOLAN	2854.5	94.375	0.553	7.097	14.583	3.195	1	0.955
3	FSiS	A1 SH	NOLAN	2855.0	89.813	0.554	7.081	14.110	2.963	1	0.932
4	FSiS	A1 SH	NOLAN	2855.5	91.563	0.560	6.733	13.189	2.979	1	0.909



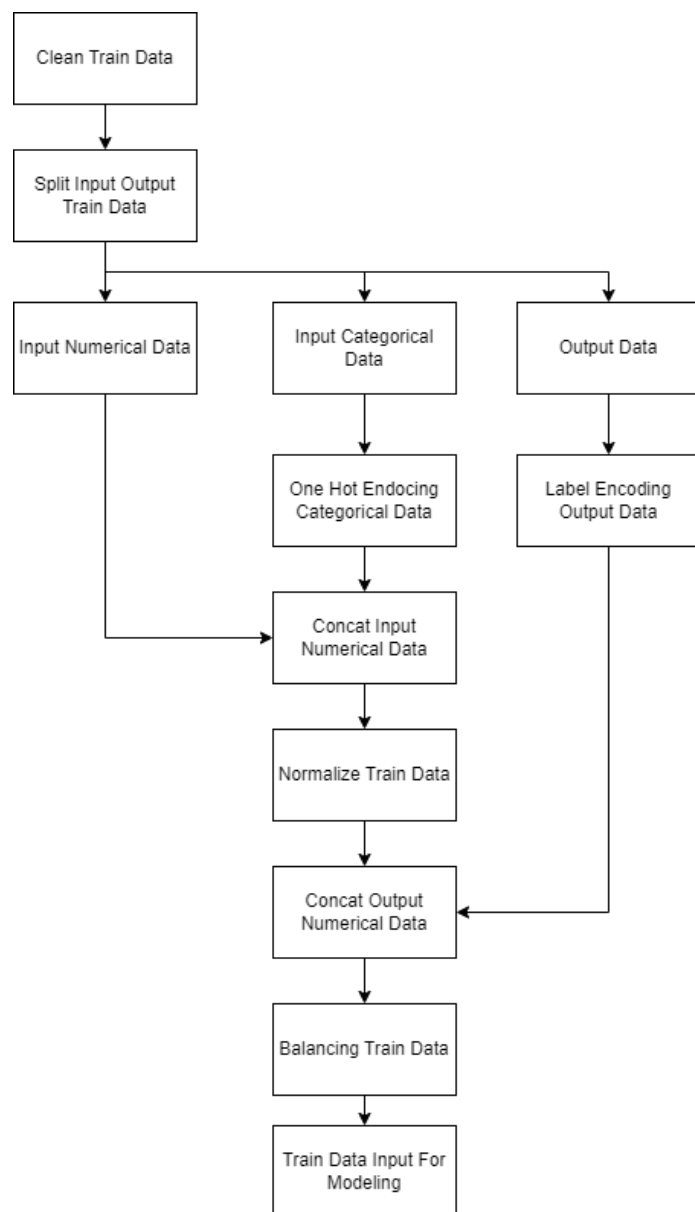
- Dataset consist of data measurement from 9 real well and 1 synthetic well (F9) to compensate category BS (Phyloid-Algae Bafflestone) in other well
- The difference on the amount of data from the real well wasn't so significant, but it is significant in the synthetic well

Data Preprocessing



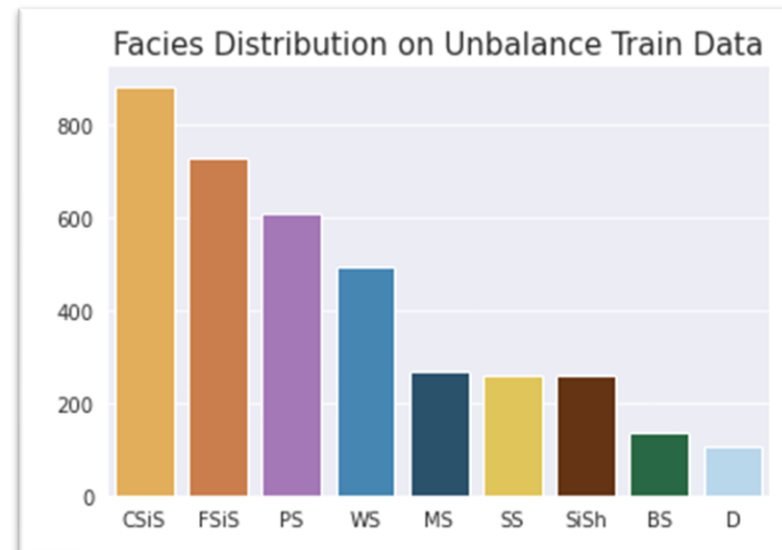
- Raw data is comprised of 10 CSV files that represent the well measurement from 10 different well
- Well 'CHURCHMAN BIBLE' was used to become the test data well, and the rest of the 9 well data serve as train data
- There are missing value in numerical data, and then it's imputed by mean value for every label categories
- Every data in train data and test data checked for data type and important range value

Feature Engineering



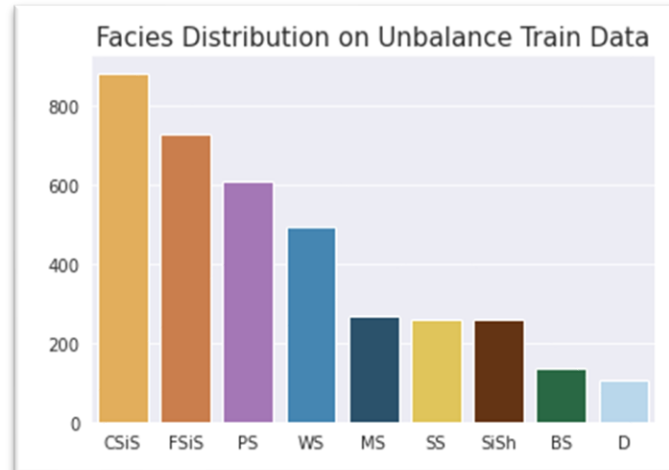
- Drop feature Formation, Well Name, Depth, and RELPOS, then split numerical, categorical, and output data
- One Hot Encoding for feature NM_M
- Label Encoding for feature output facies
- Normalize Input to have mean = 0 and standard deviation = 1
- Balancing train data using random under sampling, random over sampling, and smote

Facies	Numeric Representation
SS	0
CSiS	1
FSiS	2
SiSh	3
MS	4
WS	5
D	6
PS	7
BS	8



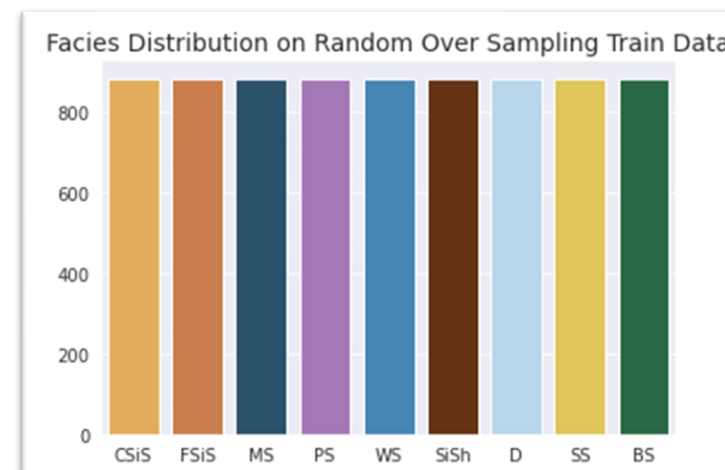
Unbalance

- 3745 data point for training
 - Facies unbalance



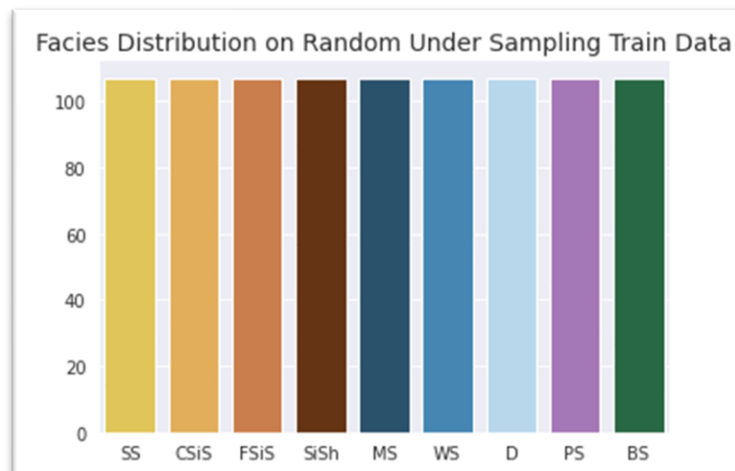
Random Over Sample

- 7956 data point for training
- Facies balance



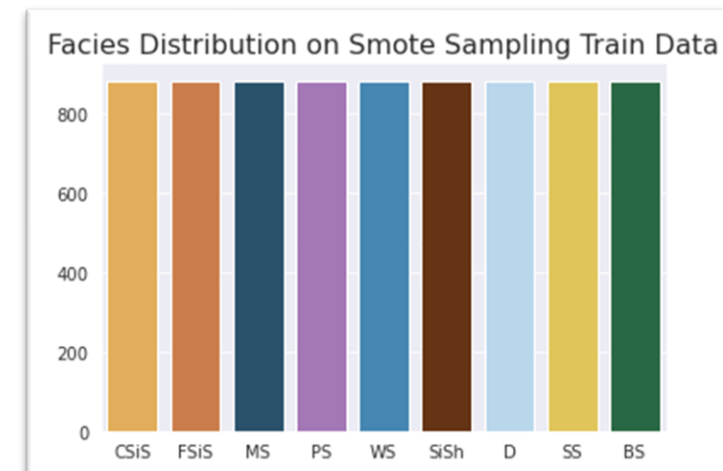
Random Under Sample

- 963 data point for training
- Facies balance

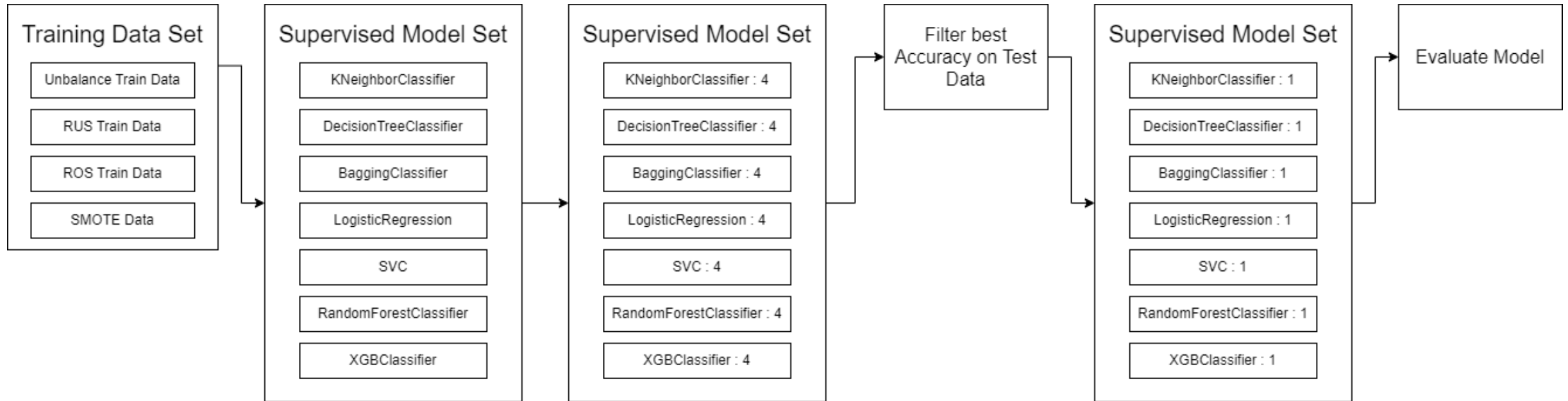


SMOTE

- 7956 data point for training
- Facies balance

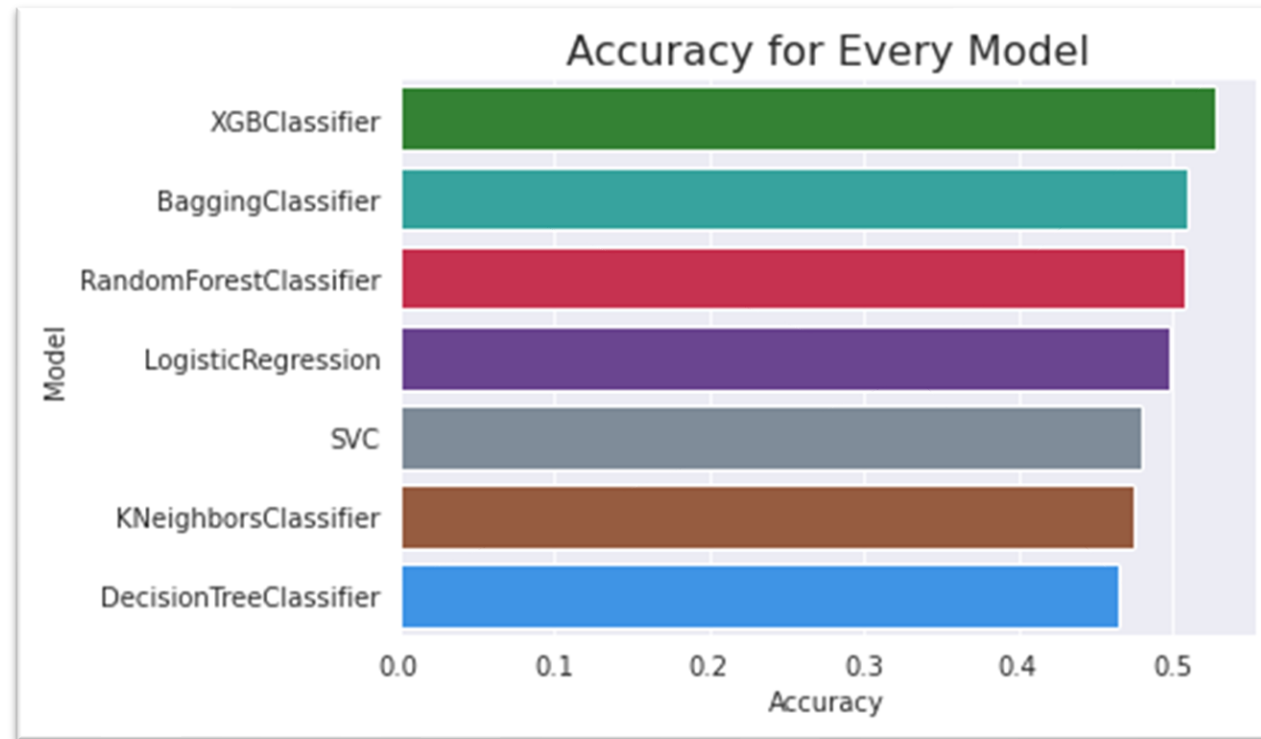


Modeling

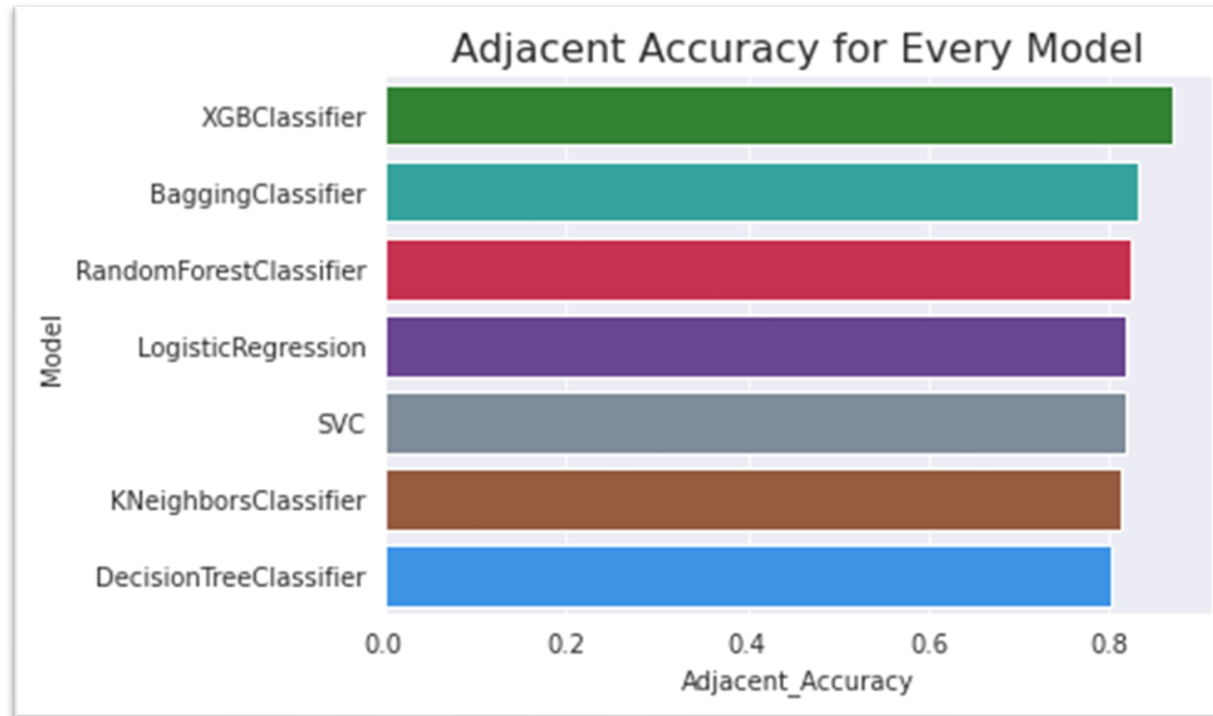


- Seven supervised model algorithm was trained by using four training data set, that will produce 28 machine learning model
- From every algorithm will be picked one with the best accuracy on data test
- From this 7 machine learning model will be picked one the best for facies classifier

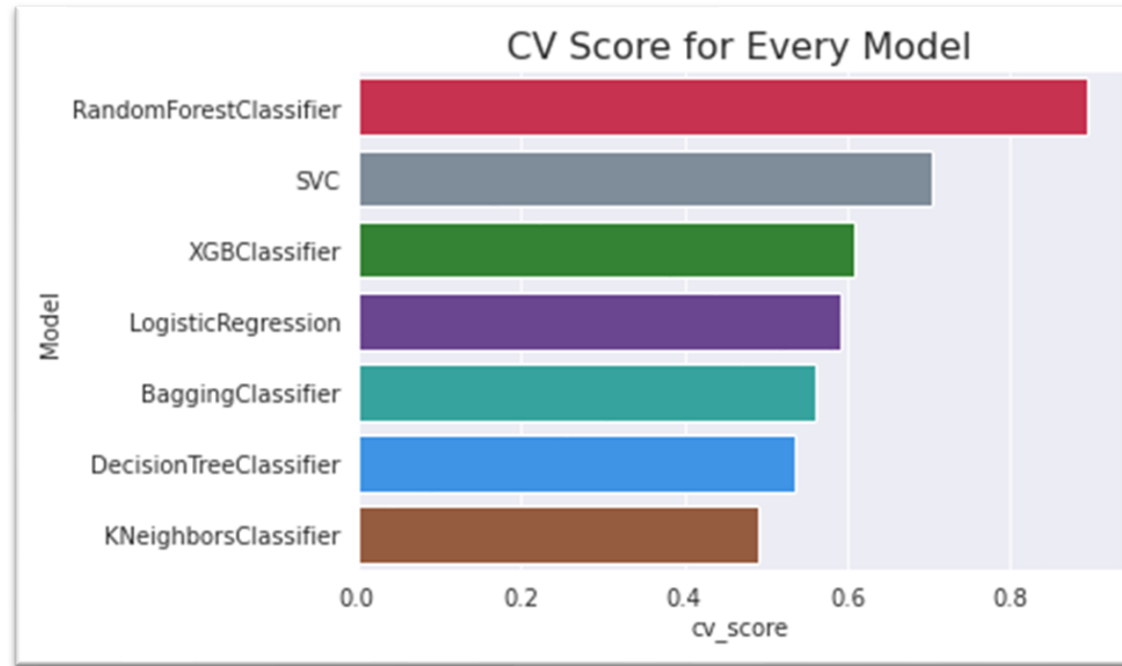
Model Evaluation



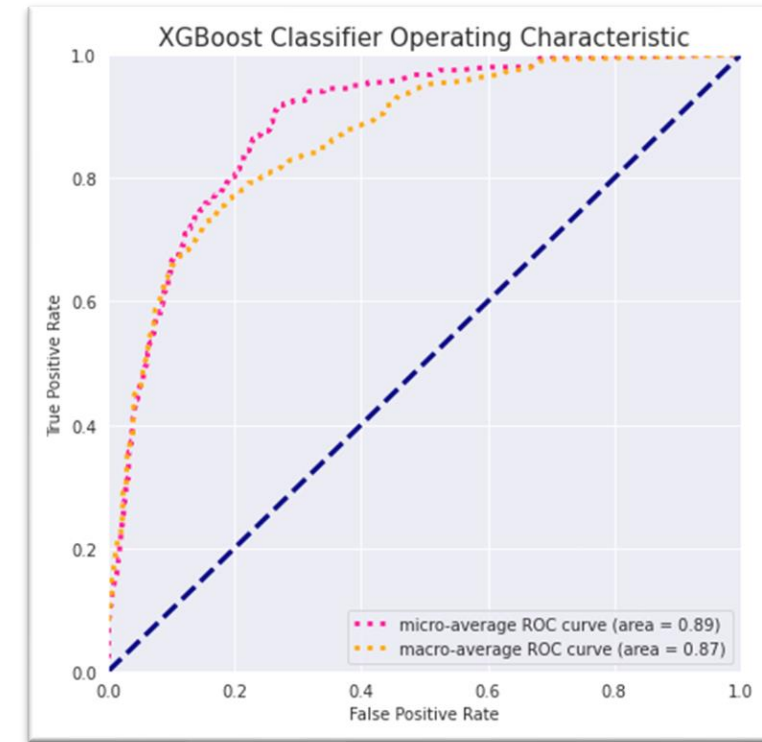
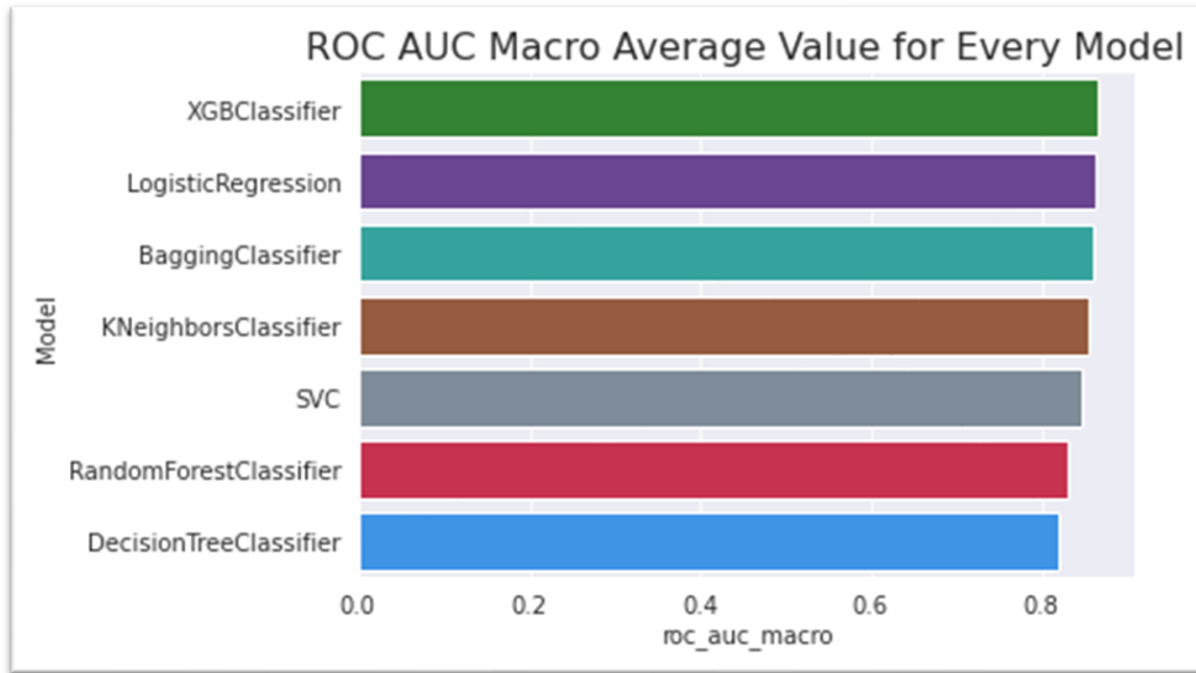
- All model have accuracy below 0.6
- XGBClassifier has the highest accuracy on test data (“CHURCHMAN BIBLE”) for 52.7% and Decision Tree Classifier has the smallest accuracy on test data for 46.5%



- All model have adjacent accuracy more than 0.8
- XGB Classifier again has the highest adjacent facies value for 86,8% and again Decision Tree Classifier become the model with the smallest adjacent accuracy on test data for 80,2%

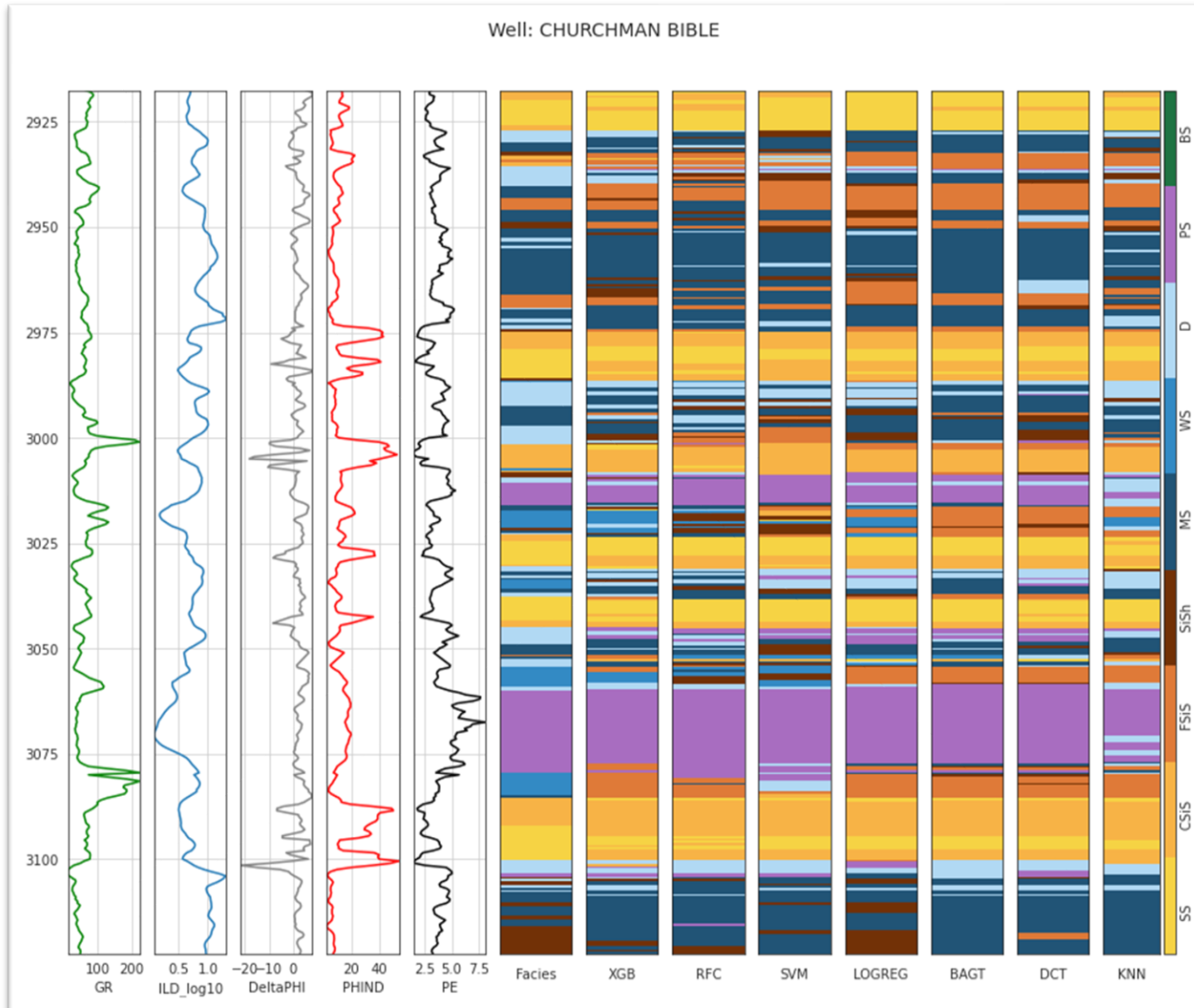


- Random Forest Classifier has the highest cv-score for 89.34%, whereas KNN has the smallest cv-score for 49,1%.
- Random forest and svc have a high difference between CV score and accuracy, we can say that for this two model is overfit on train data, even though they already pass the cross-validation process.
- For acceptable CV Score, XGB Classifier has the highest cv score for 60.85% whereas KNN has the smallest cv score for 49,1%.



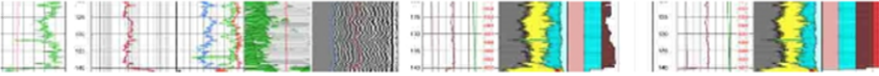
- All model have roc-auc more than 0.8
- XGB Classifier has the highest ROC-AUC score of 86.5% whereas Decision Tree Classifier has the smallest roc-auc score of 82%.
- For multi-class classification ROC-AUC curve was constructed by computing average TPR and FPR for every category.

Evaluation by Prediction Result



- every model could **predict the majority of facies** when the layer doesn't variate much, like at the depth around **3075 and 2960**
- But when it comes to variative layer like in the depth around **3050 and 3100** the predicted lithofacies become **clearly different** with actual facies.
- **XGB Classifier** with the **best performance of accuracy, adjacent accuracy, cv-score, and ROC-AUC** curve could predict the lithofacies layer better than any other model.

Front End and Back End Services



Lithofacies Classification

Enter measurement value then click predict button

1. Enter Gamma Ray Value: ⓘ

2. Enter Resistivity Value: ⓘ

3. Enter Neutron-Density Porosity Difference Value: ⓘ

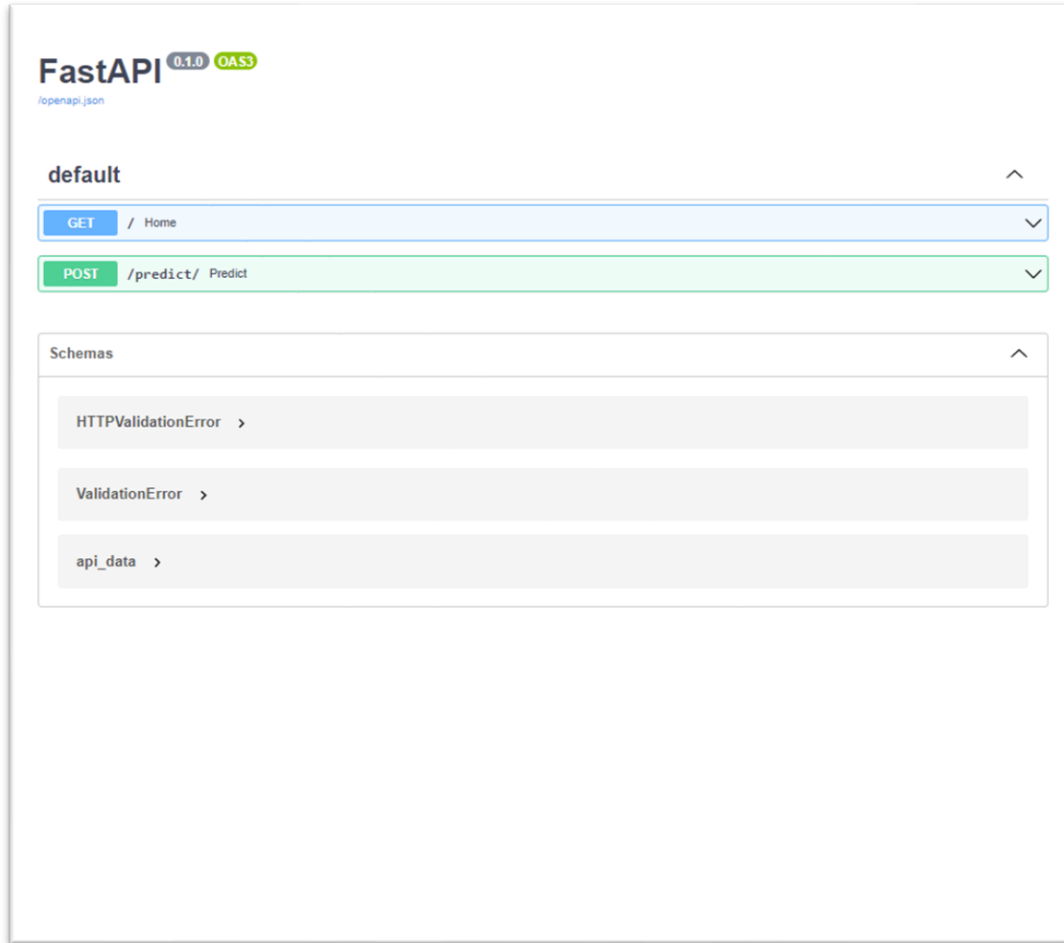
4. Enter Average Neutron-Density Porosity Value: ⓘ

5. Enter Photo-Electric Value: ⓘ

6. Enter Non-Marine Marine Value: ⓘ

Sandstone

- This project use streamlit as the front end of machine learning application
- Streamlit runs on port 8501
- Streamlit will sent api data into fast api and receive prediction result of the model



- Fast API was use as application peripheral interface for the machine learning model
- Fast API receive API data in JSON format from streamlit. Then format the data so the machine learning model could make prediction.
- After prediction Fast API will sent again the data back to streamlit for display

Pytest

```
===== test session starts =====
platform linux -- Python 3.9.12, pytest-7.2.0, pluggy-1.0.0
rootdir: /home/st_yudi/portfolio/07_Facies_Label_Deployment
plugins: anyio-3.6.2
collected 14 items

src/unit_test.py ..... [100%]

===== 14 passed in 1.08s =====
```

- Unit test was used to test every function in data_preprocessing.py and feature_engineering.py.
- This project uses the pytest library on 14 functions in both data_preprocessing.py and feature_engineering.py and all pass the unit test

Deployment

Product

Solutions

Open Source

Pricing

Search

Sign in

Sign up

StefanusYudi22 / Facies_Classification_Deployment Public

Notifications Fork 0 Star 0

<> Code Issues Pull requests **Actions** Projects Security Insights

Actions

All workflows

Python application

Management

Caches

All workflows

Showing runs from all workflows

21 workflow runs

	Event	Status	Branch	Actor
✔ Troubleshoot CICD Python application #50: Commit aff3334 pushed by StefanusYudi22		5 hours ago 3m 30s	main	...
✔ Troubleshoot CICD Python application #49: Commit acd3050 pushed by StefanusYudi22		5 hours ago 3m 10s	main	...
✖ Troubleshoot CICD Python application #48: Commit 1696eb7 pushed by StefanusYudi22		5 hours ago 2m 23s	main	...

← Python application

✔ Troubleshoot CICD #50

Summary

Jobs

Run details

docker-compose-build

docker-pull-ec2

Usage

Workflow file

Triggered via push 5 hours ago

StefanusYudi22 pushed · aff3334 main

Status

Success

Total duration

3m 30s

Artifacts

–

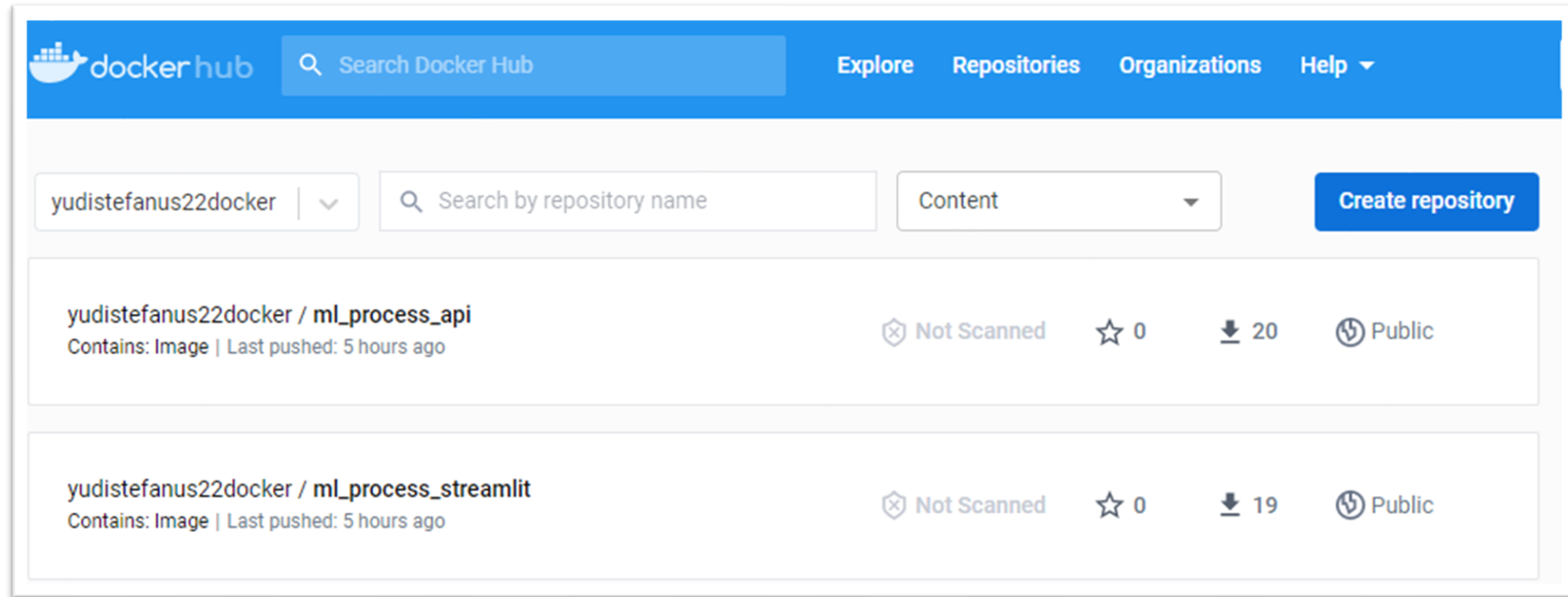
python-app.yaml

on: push

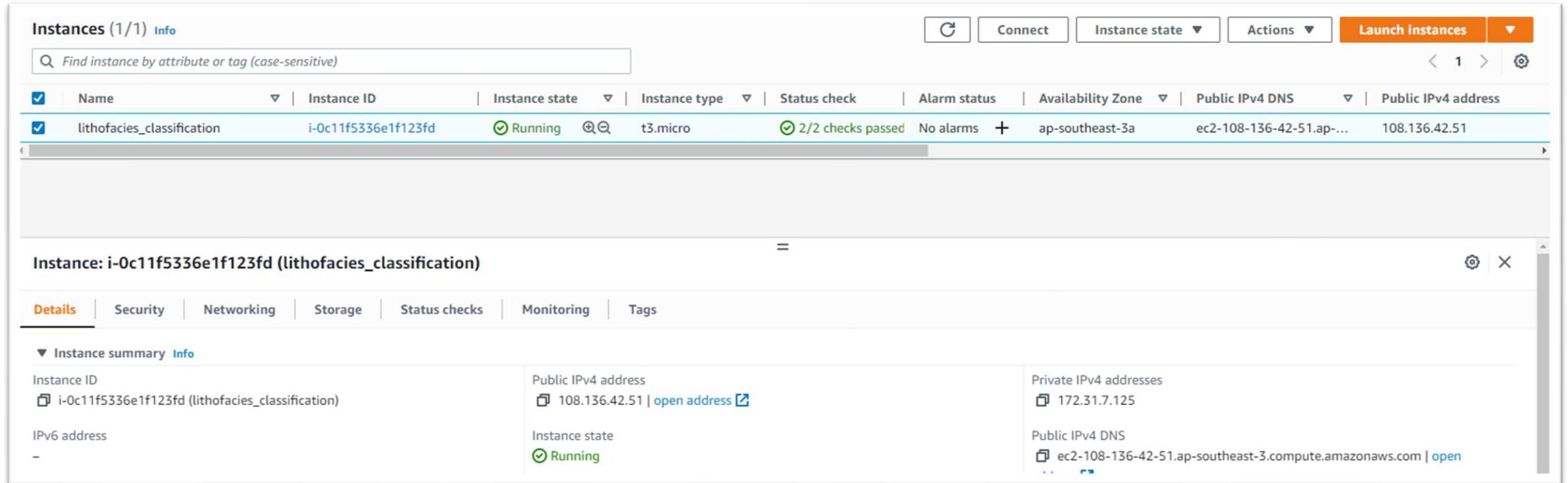
✔ docker-compose-build 2m 6s

✔ docker-pull-ec2 1m 7s

- This development use Github Action as pipeline to store the docker application in docker hub and run the ML Model in AWS Instance



- This development use docker to store the Front end and back end services
- Docker function as a container so the front-end and back-end services can be run at any operating system as long as the operation sistem have docker
- Front end and back end run on separate docker in a time



Instances (1/1) Info

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 address
lithofacies_classification	i-0c11f5336e1f123fd	Running	t3.micro	2/2 checks passed	No alarms	ap-southeast-3a	ec2-108-136-42-51.ap-...	108.136.42.51

Instance: i-0c11f5336e1f123fd (lithofacies_classification)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

▼ Instance summary Info

Instance ID i-0c11f5336e1f123fd (lithofacies_classification)	Public IPv4 address 108.136.42.51 open address	Private IPv4 addresses 172.31.7.125
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-108-136-42-51.ap-southeast-3.compute.amazonaws.com open

- This development use AWS EC2 instance as remote server for the machine learning model to run
- The IP Public for this instance is 108.136.42.51, you can access the application through port 8501 (108.136.42.51:8501)

Reference

- [1] Imamverdiyev, Y., Sukhostat, L., 2019, Lithological facies classification using deep convolutional neural network. Journal of Petroleum Science and Engineering 174 (2019) 216–228
- [2] M. Gifford, C. Agah, A., 2010, Collaborative multi-agent rock facies classification from wireline well log data, Engineering Applications of Artificial Intelligence 23 (2010) 1158–1172
- [3] W. Dunham, M. Malcolm, A. Kim Welford, J. 2020, Improved well log classification using semisupervised Gaussian mixture models and a new hyper-parameter selection strategy, Computers and Geosciences 140 (2020) 104501
- [4] W.J. Glover P., K. Mohammed-Sajed, O., Akyiiz, C., Lorinczi, P. 2022, Clustering of facies in tight carbonates using machine learning, Marine and Petroleum Geology 144 (2022) 105828
- [5] Antariksa, G. Muamar, R. Lee, J. 2022, Performance evaluation of machine learning-based classification with rock-physics analysis of geological lithofacies in Tarakan Basin, Indonesia, Journal of Petroleum Science and Engineering 208 (2022) 109250

Thank You





CERTIFICATE OF COMPLETION

Has Been Awarded To

Stefanus Yudi Irwan

for successfully completing in class

**Introduction to Machine Learning - I & Introduction to
Machine Learning - II**

October 11, 2022 - December 11, 2022



Adityo Sanjaya
CEO Pacmann

SIGNATURE: 19/12/2022



Verifikasi Sertifikat
<https://sertifikat.pacmann.ai/ZM13VrQYRa0mTMN>



CERTIFICATE OF COMPLETION

Has Been Awarded To

Stefanus Yudi Irwan

for successfully completing in class

Machine Learning Process - I & Machine Learning Process -

II

October 10, 2022 - December 11, 2022



Adityo Sanjaya
CEO Pacmann

SIGNATURE: 19/12/2022



Verifikasi Sertifikat
<https://sertifikat.pacmann.ai/jhAyjQ2ykmhJSVd>

Reach me ! For discussion



[My-Resume](#)

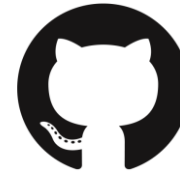


[My-Email](#)



[My-LinkedIn](#)

Project Repository & Presentation



[Project-Repository](#)



[YouTube Presentation](#)