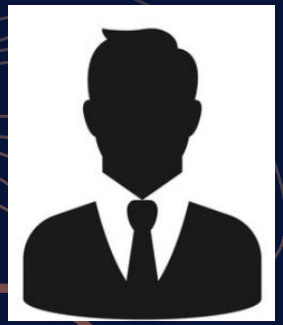
FOREST FIRES
Classification with Deep Learning

By: Team Alpha Batch 8

Team Member

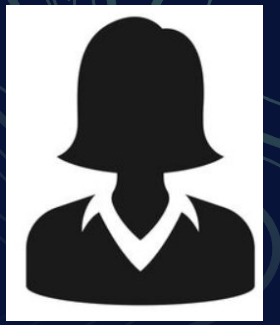

Azka


Rafi


Farell


Devon


Novie


Karimah

Forest Fires Task

Classification

Prediksi Potensi Kebakaran Hutan

Request #1

Tuning Hyperparamater Deep Learning

Request #2

Performance Improvement

TABLE OF CONTENTS

01 Exploratory Data Analysis

Data Shape, Missing and Duplicate

Data Dimension

517 Total records and 13 Features

Missing Values

There is no missing values

Duplicate Data

Only containing 4 duplicate data

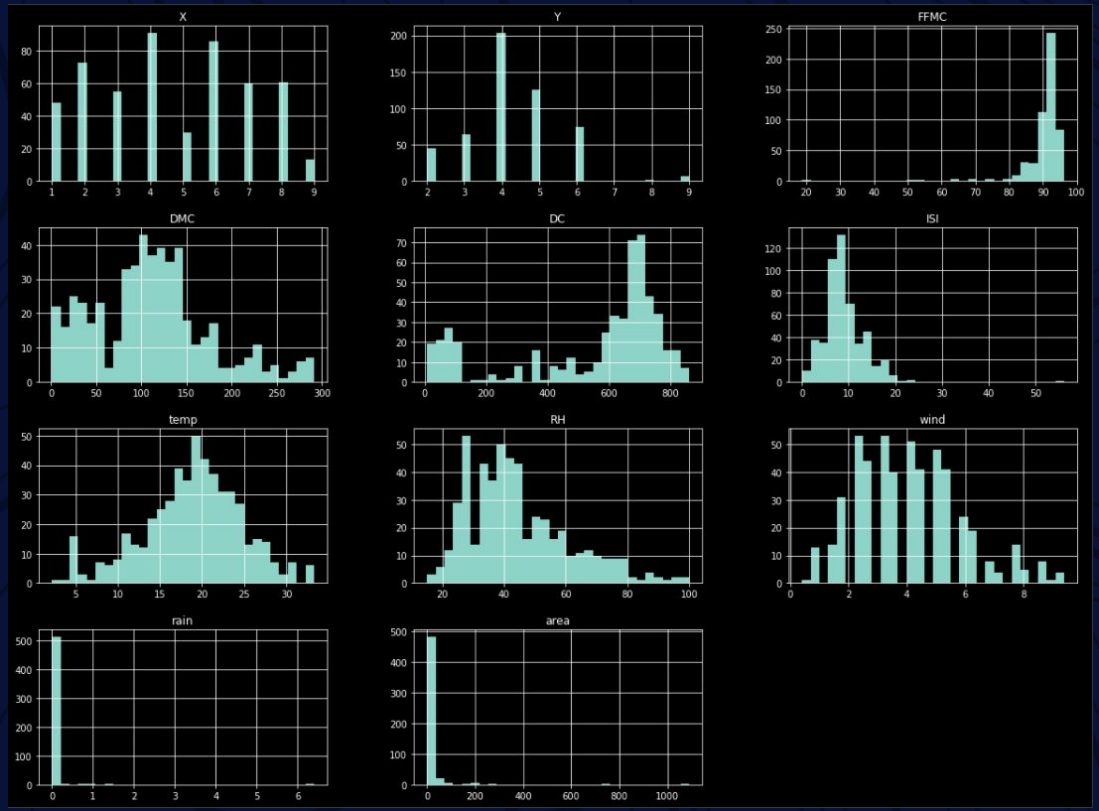
Data Summary Statistics

	count	mean	std	min	25%	50%	75%	max
X	517.0	4.669246	2.313778	1.0	3.0	4.00	7.00	9.00
Y	517.0	4.299807	1.229900	2.0	4.0	4.00	5.00	9.00
FFMC	517.0	90.644681	5.520111	18.7	90.2	91.60	92.90	96.20
DMC	517.0	110.872340	64.046482	1.1	68.6	108.30	142.40	291.30
DC	517.0	547.940039	248.066192	7.9	437.7	664.20	713.90	860.60
ISI	517.0	9.021663	4.559477	0.0	6.5	8.40	10.80	56.10
temp	517.0	18.889168	5.806625	2.2	15.5	19.30	22.80	33.30
RH	517.0	44.288201	16.317469	15.0	33.0	42.00	53.00	100.00
wind	517.0	4.017602	1.791653	0.4	2.7	4.00	4.90	9.40
rain	517.0	0.021663	0.295959	0.0	0.0	0.00	0.00	6.40
area	517.0	12.847292	63.655818	0.0	0.0	0.52	6.57	1090.84

Data Visualization

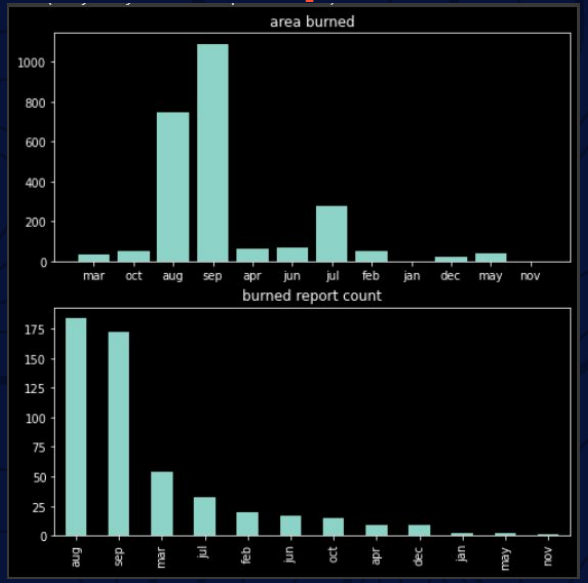
Distribution Plot

Attribute Distribution Plot



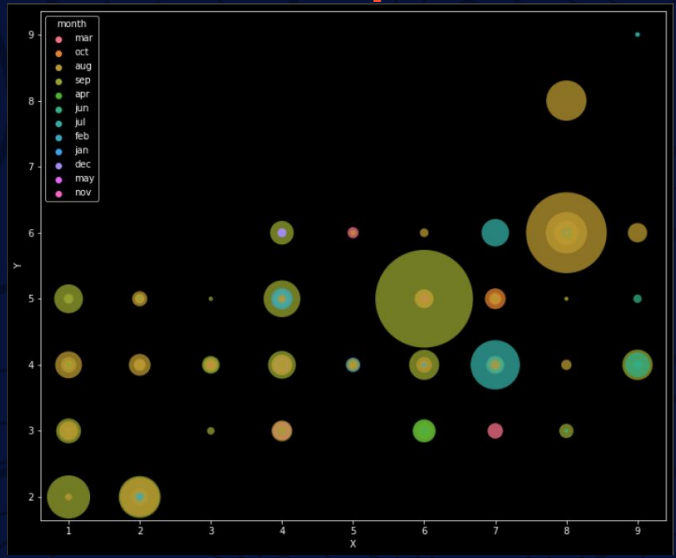
Data Visualization

Histplot



Burned Area & Burned Report Count each Month

Scatterplot

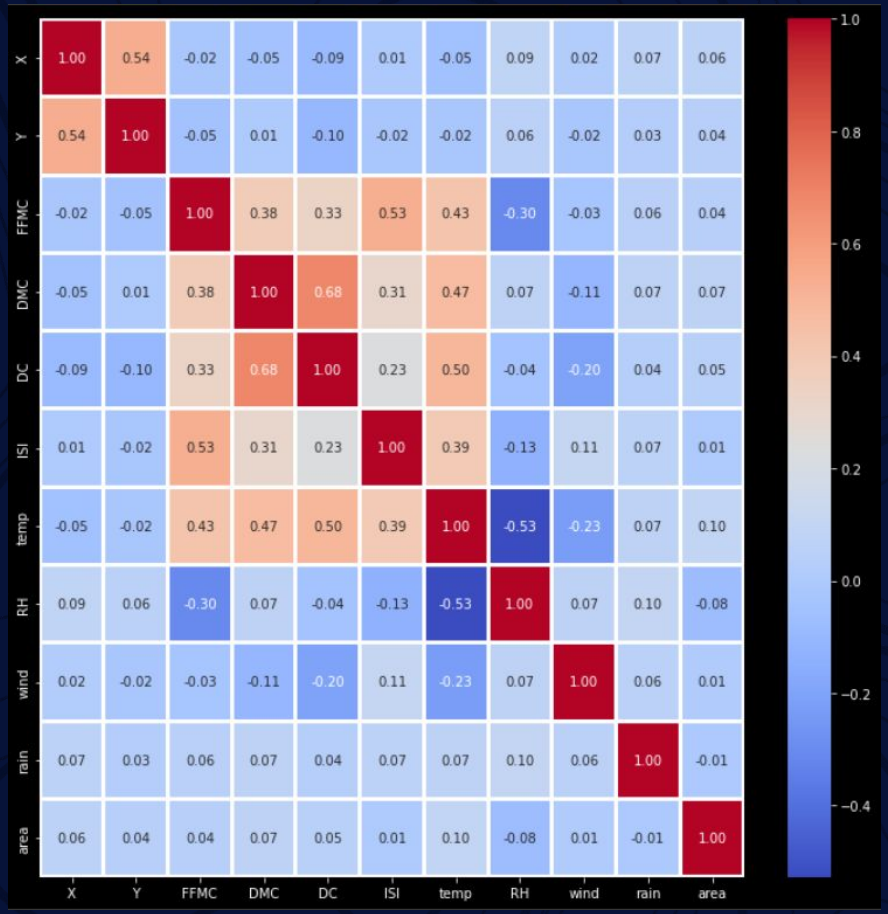


Scatter Plot of Burned Area Coordinates

Data Visualization

Heatmap

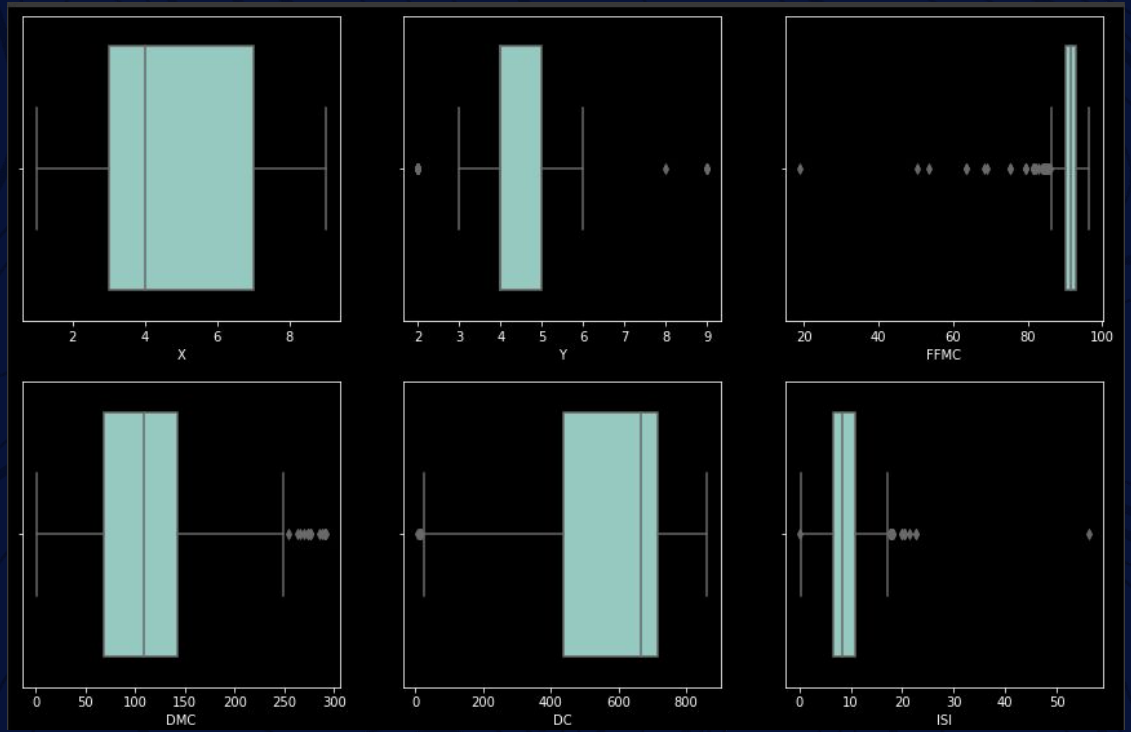
Heatmap of Linear Correlation of each Attribute



Data Visualization

Boxplot

Visualize Outliers in Dataset



02 Preprocessing

Categorizing Severity of Forest Fires

Skenario 1

Membagi label menjadi 2 kategori yaitu apakah arena tersebut Terbakar atau Tidak Terbakar.

Skenario 2

Membagi label menjadi 5 kategori tingkat kerusakan kebakaran hutan berdasarkan luas area terbakar.

Encoding & Features Selection

Label Encoding

Melakukan label encoding pada atribut “month” menjadi numerical values untuk mempermudah proses komputasi

Drop “DC”

Melakukan drop terhadap atribut “DC” untuk menghindari multicollinearity

Drop “Day”

Drop atribut “day” dilakukan karena atribut tersebut memiliki distribusi yang rata terhadap data target

03 Data Preparation

Scaling & Splitting Data

Scaling Data

Data scaling dilakukan menggunakan metode standarisasi untuk menyamakan range antar data agar mempermudah proses komputasi

Splitting Data

Memisahkan antara data train dan data test dengan perbandingan 70:30 sebelum dimasukkan ke dalam model

04 Modelling & Evaluate

1

Skenario 1

Data scaling dilakukan menggunakan metode standarisasi untuk menyamakan range antar data agar mempermudah proses komputasi

Build Model

Epoch

Using 400 epoch

Layers

5 layers are used in the model

Activation Function

Using RELU for each hidden layers and Softmax for output layer

Optimizer

Using SGD optimizer with 0.001 of learning rate and 0.9 momentum

Loss Function

Using binary crossentropy as loss function

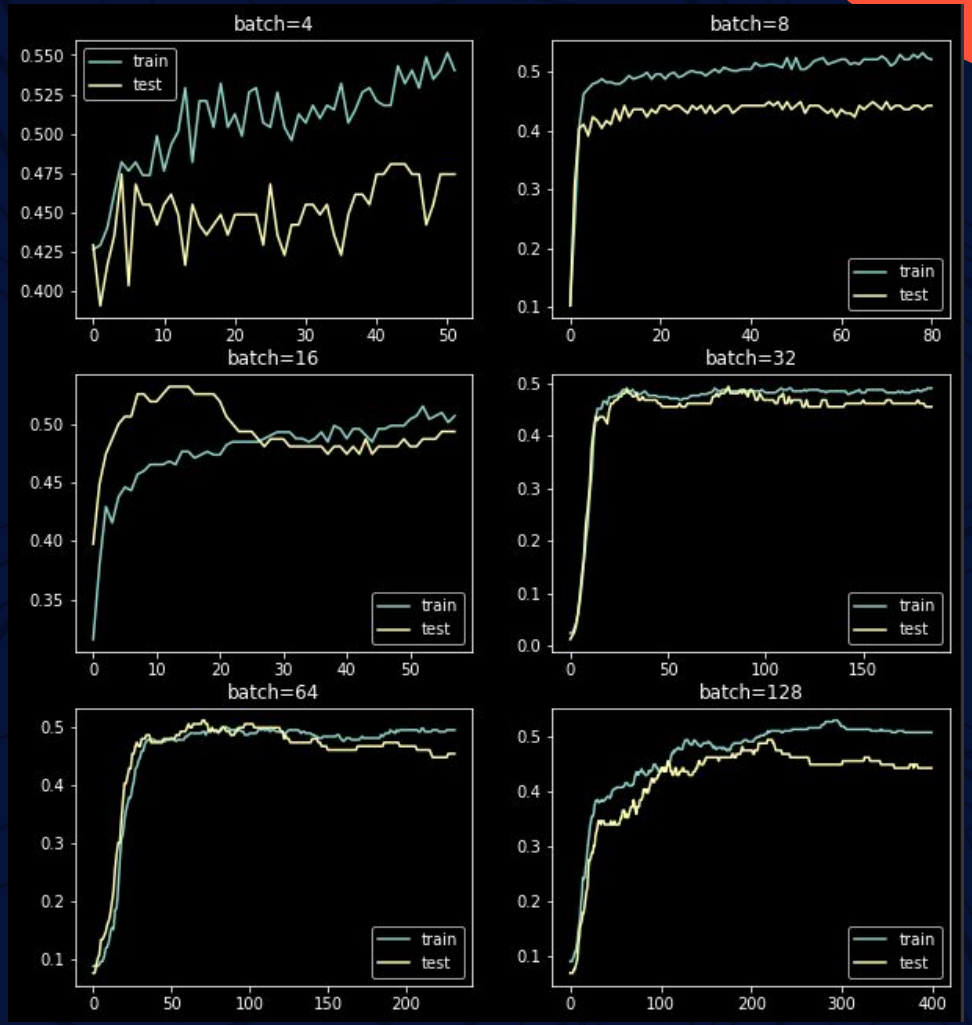
Batch Size

[4, 8, 16, 32, 64, 128]

Early Stop

Using early stop as performance improvement method with patience = 10

Model Result



2

Skenario 2

Membagi label menjadi 5 kategori tingkat kerusakan kebakaran hutan berdasarkan luas area terbakar.

Build Model

Epoch

Using 400 epoch

Layers

5 layers are used in the model

Activation Function

Using RELU for each hidden layers and Softmax for output layer

Optimizer

Using SGD optimizer with 0.001 of learning rate and 0.9 momentum

Loss Function

Using binary crossentropy as loss function

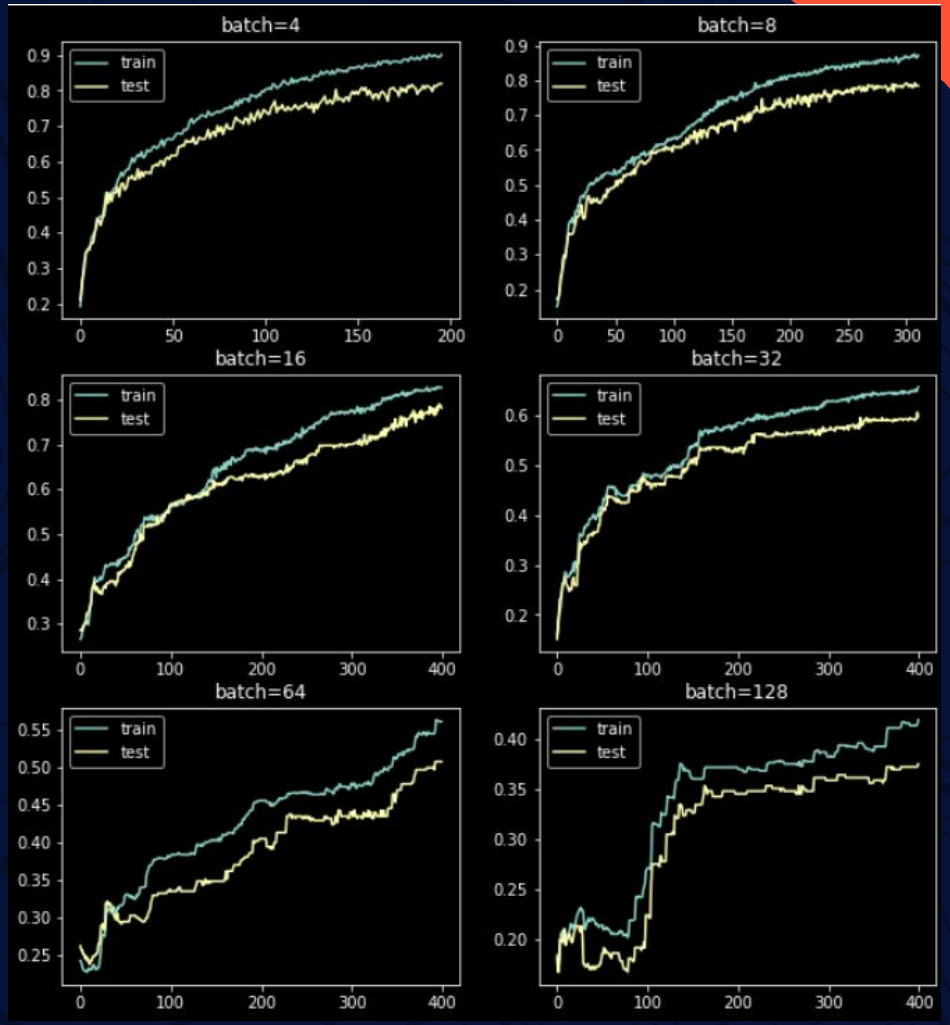
Batch Size

[4, 8, 16, 32, 64, 128]

Early Stop

Using early stop as performance improvement method with patience = 10

Model Result



3 Perbandingan

Membandingkan dengan algoritma selain deep-learning

Decision Tree

```python
#Decision tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

dtc= DecisionTreeClassifier(criterion="gini", max_depth=4)
dtc.fit(X_train, y_train1)

ypred= dtc.predict(X_train, check_input=True)
print("train acc:", accuracy_score(y_train1, ypred))

ypred= dtc.predict(X_test, check_input=True)
print("validation acc: ", accuracy_score(y_test1, ypred))
```

```
train acc: 0.6897506925207756
validation acc:  0.5512820512820513
```

THANK YOU