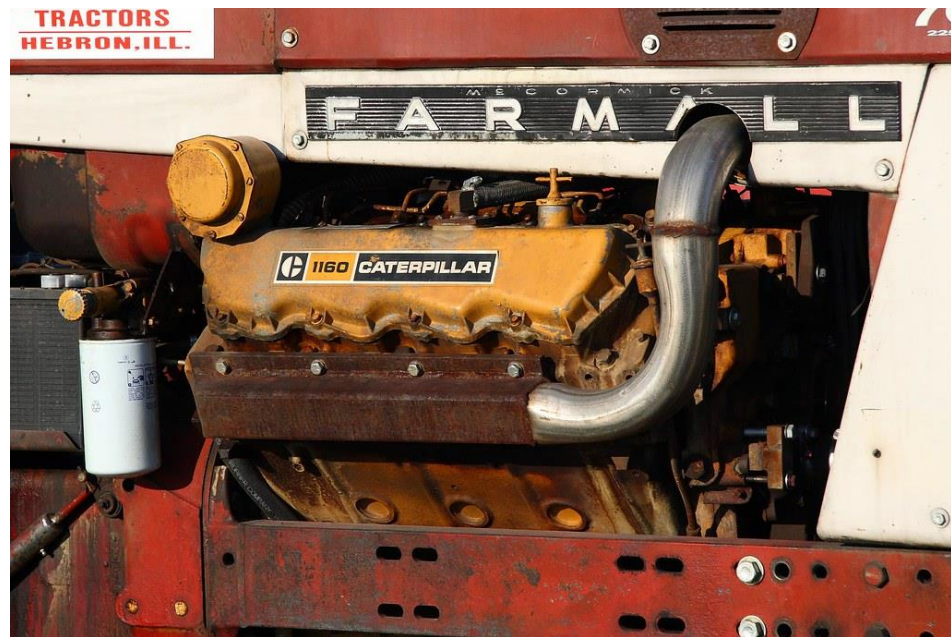


# CAPSTONE DATA MINING PROJECT



AUG 01, 2021

TEAM 404

**PRESENTED BY:**

SAMYUKTHA M S  
NITHIIN KATHIRESAN  
JEYAM PALANIAPPAN

CSE 2<sup>ND</sup> YEAR  
PSG COLLEGE OF TECHNOLOGY, COIMBATORE

# CAPSTONE DATA MINING PROJECT

TEAM 404

## INTRODUCTION

Cat Digital uses telematics data collected from customers' equipment to proactively detect potential present or future equipment failures. In this project we have developed an AI solution that reproduces and possibly surpasses a human expert's thought processes in predicting equipment failures using telematics data (specifically Cat® Product Link).

## PROJECT ABSTRACT

Using the telematics data collected, the model predicts when the threshold value of a particular event will occur for a given candidate and alerts the customer beforehand. As a future development, we have also planned to incorporate certain features such as a mail to the customer 3-5 days before warning about the system failure. Also, a stock checking in the nearest store so that the customer can easily replace or repair the faulted equipment.

## HOW THE MODEL WORKS

We have used GOOGLE COLAB to build our model, Initially the dataset is cleaned and splitted according to the candidate's name into multiple CSV files. Then each candidate's csv files are further splitted according to the event's name into CSV files. In this way we have splitted the entire dataset according to our need into CSV files that contains a particular event only of a particular candidate. The model uses linear regression to calculate the number of days left before the equipment fails due to a particular event given the input of a candidate name and the current date.

## LANGUAGE AND LIBRARIES USED

- PYTHON
- PANDAS
- SCIKIT
- NUMPY
- MATPLOTLIB

### CODE





















```
import pandas as pd
data = pd.read_csv("/content/catathon/CAT_Training_Dataset_V3 File.csv")
for (candidate), group in data.groupby(['candidate']):
    group.to_csv(f'{candidate}.csv', index=False)
```

#THE GIVEN DATASET IS SPLITTED ACCORDING TO THE CANDIDATE

```
import pandas as pd
import os
import glob
path = os.getcwd()
csv_files = glob.glob(os.path.join(path, "*.csv"))
for f in csv_files:
    df = pd.read_csv(f)
    for (event,candidate), group in data.groupby(['event','candidate']):
        group.to_csv(f'{candidate}{event}.csv', index=False)
```

#THE FILES ARE FURTHER SPLITTED ACCORDING TO THE EVENT NAME

### SNAPSHOT OF THE OUTPUT

-  ABCDE00045Component1\_SensorJ\_High.csv
-  ABCDE00045Component1\_SensorJ\_exceededLimit.csv
-  ABCDE00045Component20\_maxSensorC\_High.csv
-  ABCDE00045Component21\_maxSensorC\_High.csv
-  ABCDE00045Component22\_maxSensorC\_High.csv
-  ABCDE00045Component23\_FluidB\_SensorA\_High.csv
-  ABCDE00045Component24\_maxSensorC\_High.csv
-  ABCDE00045Component27\_SensorA\_High.csv
-  ABCDE00045Component28\_maxSensorC\_High.csv
-  ABCDE00045Component31\_maxSensorC\_High.csv
-  ABCDE00045Component32\_FluidA\_SensorA\_High.csv
-  ABCDE00045Component3\_deltaSensorB\_High.csv
-  ABCDE00045Component6\_deltaSensorB\_High.csv
-  ABCDE00045Component9\_deltaSensorB\_High.csv
-  ABCDE00045Secondary\_Component32\_Fluid\_SensorA\_High.csv
-  ABCDE00046.csv
-  ABCDE00046Component12\_FluidC\_SensorB\_Low.csv
-  ABCDE00046Component13\_SensorB\_Low.csv
-  ABCDE00046Component14\_discreteSensorD\_Low.csv
-  ABCDE00046Component15\_discreteSensorE\_Up.csv

1 to 11 of 11 entries						Filter
ID	candidate	date	units	event	occur_count	svrty_level
59585	ABCDE00048	28334	409.0	Component25_discreteaSensorE_Limit	1	2
59593	ABCDE00048	28360	613.0	Component25_discreteaSensorE_Limit	1	2
59699	ABCDE00048	28544	2082.0	Component25_discreteaSensorE_Limit	1	2
60089	ABCDE00048	30248	5248.0	Component25_discreteaSensorE_Limit	1	2
60375	ABCDE00048	30468	6709.0	Component25_discreteaSensorE_Limit	1	2
60390	ABCDE00048	30472	6733.0	Component25_discreteaSensorE_Limit	1	2
60420	ABCDE00048	30506	6938.0	Component25_discreteaSensorE_Limit	1	2
61089	ABCDE00048	34254	11491.0	Component25_discreteaSensorE_Limit	1	2
61090	ABCDE00048	34254	11491.0	Component25_discreteaSensorE_Limit	1	2
61138	ABCDE00048	34270	11584.0	Component25_discreteaSensorE_Limit	1	2
61275	ABCDE00048	34328	11920.0	Component25_discreteaSensorE_Limit	1	2

Show 25 per page

```

candidate=input("Enter the candiddate name:")
path="/content/"+candidate+".csv"
import pandas as pd
df = pd.read_csv (path)
df.head
eventList=df.event.unique()
print("\nNumber of events in this candidate:",len(eventList))
for flag in eventList:
    print('\n',flag)
    path1="/content/"+candidate+flag+".csv"
    df1=pd.read_csv(path1)
    c=df1["units"]
    maxValue=c.max()
    thr=maxValue+5000
    print("The Threshold Value is",thr,'\n')
    import matplotlib.pyplot as plt
    import csv
    if df1.shape[0]<3:
        print("Model can't be executed")
    else:
        x = []
        y = []
        X = []
        with open(path1,'r') as csvfile:
            lines = csv.reader(csvfile, delimiter=',')
            for row in lines:
                if row[3]!="units":
                    X.append([float(row[3])])
                    x.append(float(row[3]))
                    y.append(int(row[2]))

```

## CAPSTONE DATA MINING PROJECT

---

```
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

x_train , x_val , y_train , y_val = train_test_split(X,y)

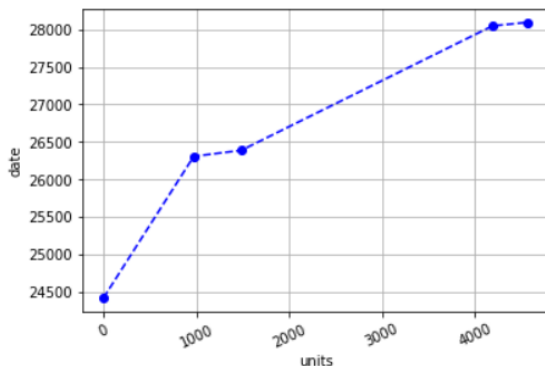
plt.plot(x, y, color = 'b', linestyle = 'dashed',marker = 'o')
plt.xticks(rotation = 25)
plt.xlabel('units')
plt.ylabel('date')
plt.grid()
plt.show()
import numpy as np
from sklearn.linear_model import LinearRegression
new_model= LinearRegression()
new_model.fit(X,y)
preds=new_model.predict([[thr]])
print(int((preds)/1440),"days from the date of reference")
new_model.fit(x_train,y_train)
preds=new_model.predict(x_val)
mae=mean_absolute_error(y_val,preds)
print("Mean Absolute Error",round((mae)/1440,2),"days")
```

## SNAPSHOT OF THE OUTPUT

Enter the candiddate name:ABCDE00049

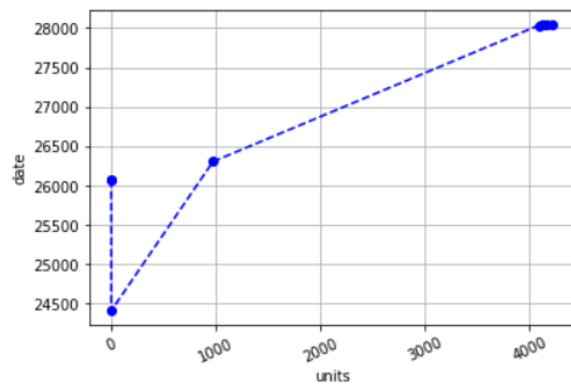
Number of events in this candidate: 22

Component8\_SensorB\_Low  
The Threshold Value is 9573.0



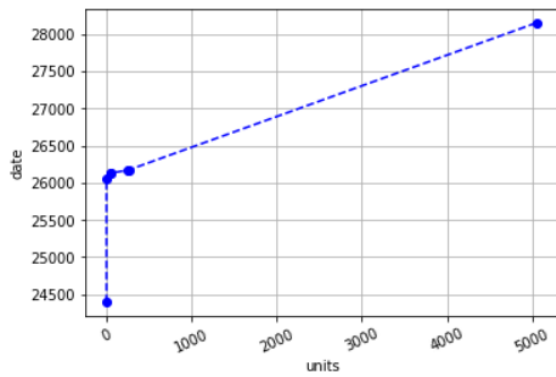
22 days from the date of reference  
Mean Absolute Error 0.25 days

Component13\_SensorB\_Low  
The Threshold Value is 9221.0



21 days from the date of reference  
Mean Absolute Error 0.08 days

Component4\_SensorB\_discreteSensorD\_Low  
The Threshold Value is 10062.0

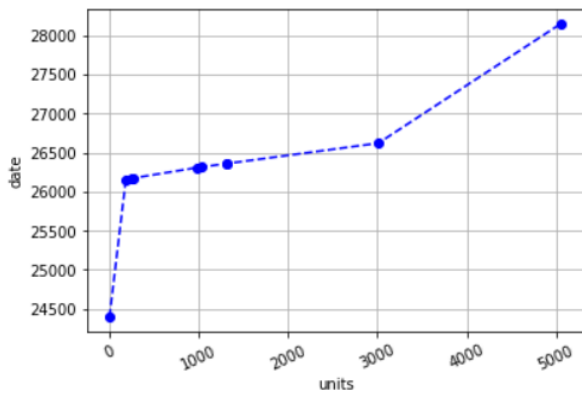


21 days from the date of reference  
Mean Absolute Error 0.59 days

Component29\_deltaSensorG\_High  
The Threshold Value is 5000.0

Model can't be executed

Component5\_SensorB\_discreteSensorD\_Low  
The Threshold Value is 10062.0



21 days from the date of reference  
Mean Absolute Error 0.68 days

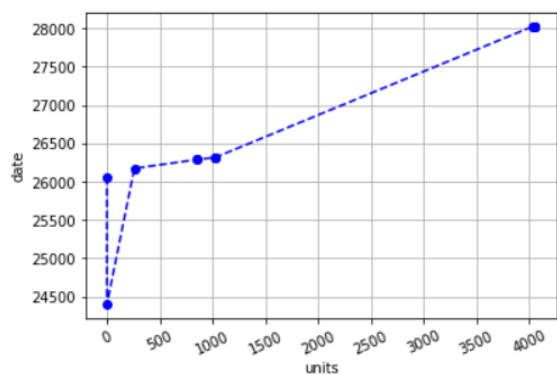
Component19\_FluidE\_discreteSensorD\_Low  
The Threshold Value is 5000.0

Model can't be executed

## CAPSTONE DATA MINING PROJECT

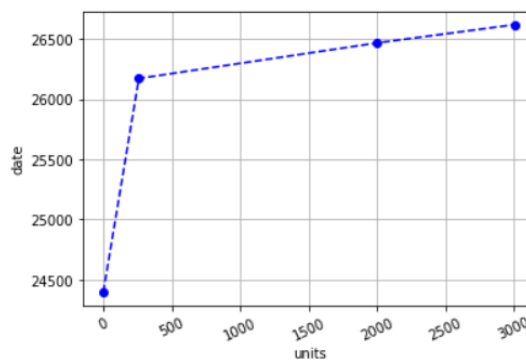
---

Component14\_discreteSensorD\_Low  
The Threshold Value is 9056.0



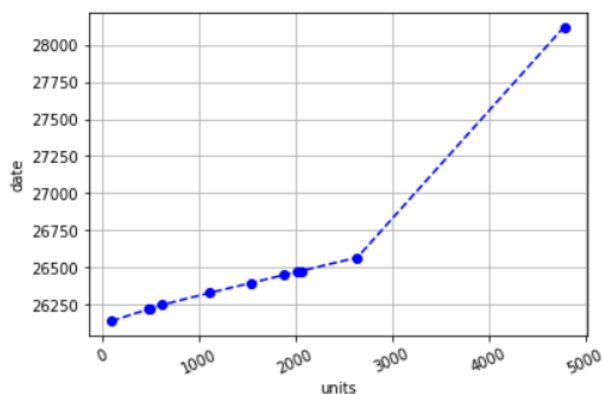
21 days from the date of reference  
Mean Absolute Error 0.18 days

Component17\_SensorB\_discreteSensorD\_Low  
The Threshold Value is 8011.0



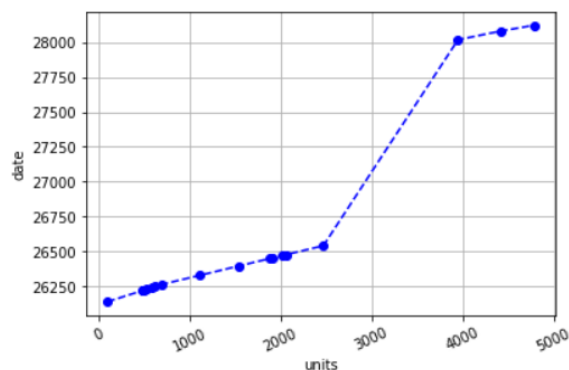
20 days from the date of reference  
Mean Absolute Error 1.21 days

Component16\_FluidB\_SensorA\_High  
The Threshold Value is 9783.0



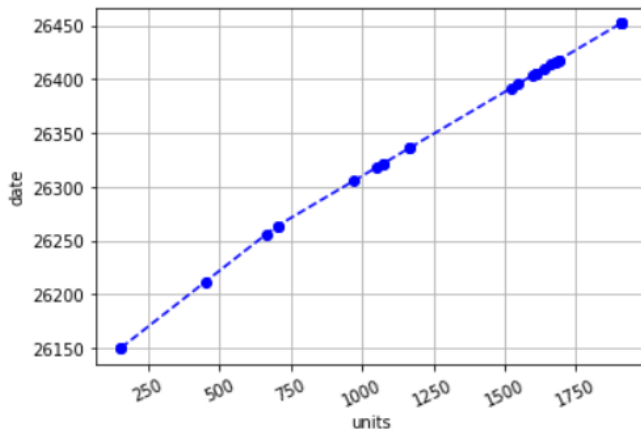
20 days from the date of reference  
Mean Absolute Error 0.28 days

Component23\_FluidB\_SensorA\_High  
The Threshold Value is 9783.0



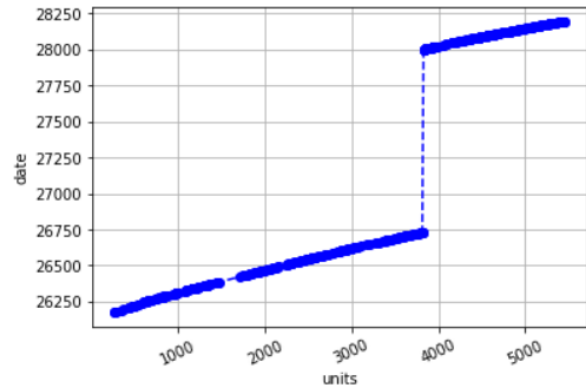
20 days from the date of reference  
Mean Absolute Error 0.12 days

Component27\_SensorA\_High  
The Threshold Value is 6906.0



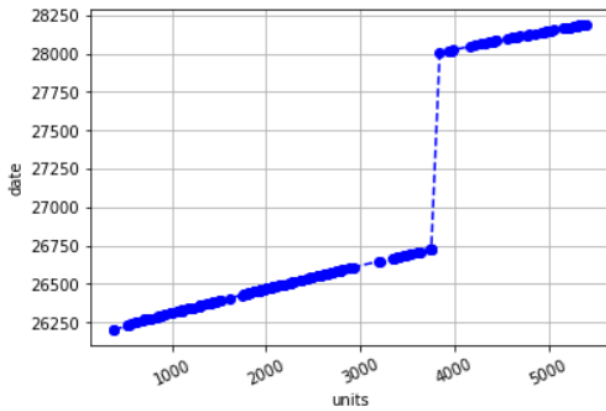
18 days from the date of reference  
Mean Absolute Error 0.0 days

Component15\_discreteSensorE\_Up  
The Threshold Value is 10446.0



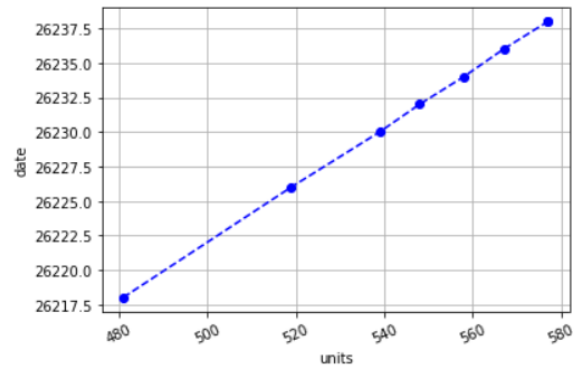
21 days from the date of reference  
Mean Absolute Error 0.19 days

Component1\_SensorJ\_exceededLimit  
The Threshold Value is 10394.0



21 days from the date of reference  
Mean Absolute Error 0.13 days

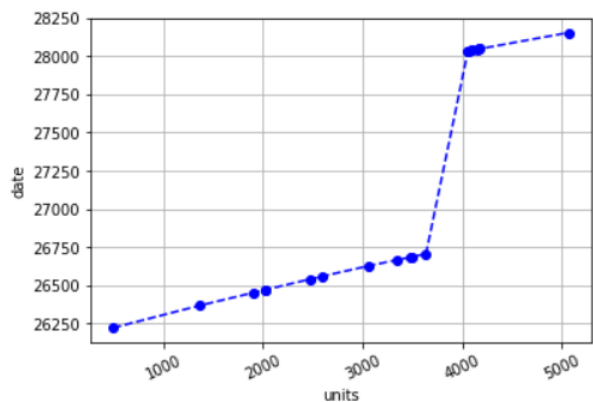
Component3\_deltaSensorB\_High  
The Threshold Value is 5577.0



18 days from the date of reference  
Mean Absolute Error 0.0 days

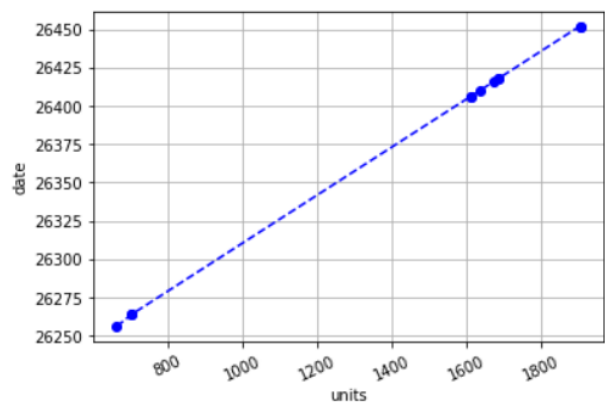


A\_Component10\_FluidB\_SensorA\_High  
The Threshold Value is 10062.0



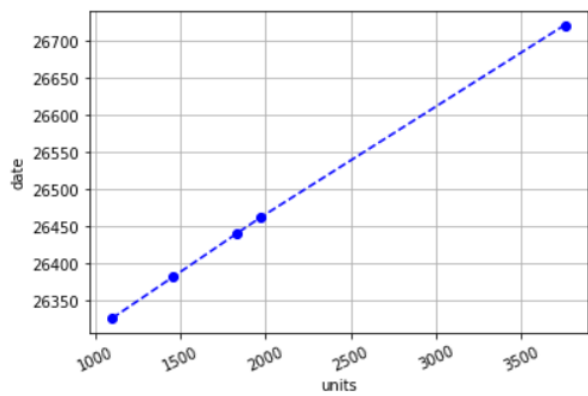
21 days from the date of reference  
Mean Absolute Error 0.18 days

Component1\_SensorG\_Limit  
The Threshold Value is 6906.0



18 days from the date of reference  
Mean Absolute Error 0.0 days

Component1\_SensorJ\_High  
The Threshold Value is 8755.0

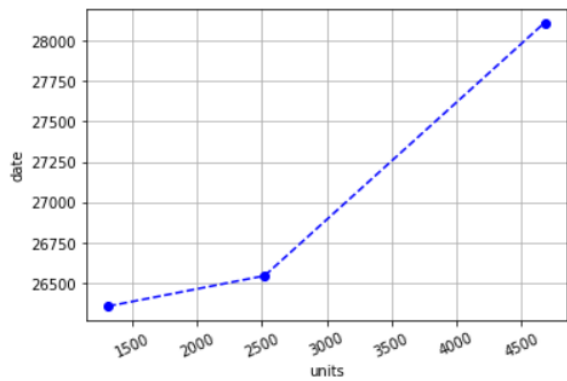


19 days from the date of reference  
Mean Absolute Error 0.01 days

Component12\_FluidC\_SensorB\_Low  
The Threshold Value is 7695.0

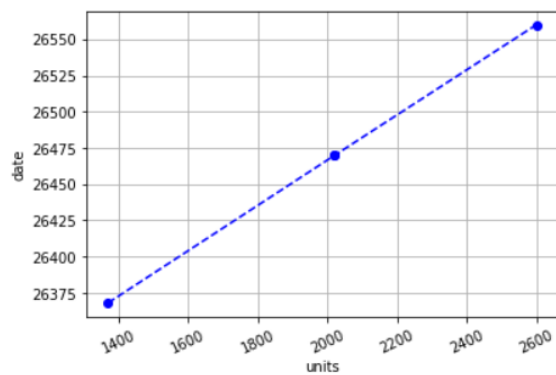
Model can't be executed

Component25\_discreteaSensorE\_Limit  
The Threshold Value is 9678.0



21 days from the date of reference  
Mean Absolute Error 0.3 days

B\_Component10\_FluidB\_SensorA\_High  
The Threshold Value is 7600.0



18 days from the date of reference  
Mean Absolute Error 0.0 days

Component2\_discreteSensorH\_Up  
The Threshold Value is 8242.0

Model can't be executed

Component6\_deltaSensorB\_High  
The Threshold Value is 9011.0

Model can't be executed

## **SALIENT FEATURES OF THE MODEL**

- THE MODEL TAKES CARE OF ERRONEOUS DATA EFFICIENTLY, SAY IF THERE IS ANY DISCREPANCY IN THE TYPE OF DATA IT IS AUTOMATICALLY OMITTED.
- THE MEAN ABSOLUTE ERROR ON AN AVERAGE IS ABOUT 4 HOURS WHICH CAN BE MADE MORE ACCURATE BY INCREASING THE SIZE OF THE DATASET.
- FOR SOME EVENTS WHERE THERE ISN'T ENOUGH DATA TO PREDICT THE FAILURE DATE, THE MODEL AUTOMATICALLY HANDLES THE ERROR.

## **FUTURE SCOPE AND DEVELOPMENT**

- USE OF IOT IN PREVENTING EQUIPMENT FAILURE
- USING AI AND PYTHON AUTOMATION TO PREVENT FAILURES
- MODEL CAN UPSCALED USING CLOUD SERVICE
- ONCE THE DATE OF FAILURE OF THE PARTICULAR COMPONENT IS PREDICTED,WE CAN NOTIFY THE CUSTOMER ABOUT THE AVAILABILITY OF THAT COMPONENT IN THE NEAREST STORE POSSIBLE
- THE ACCURACY CAN BE IMPROVED BY EXPANDING THE DATASET

- THE ENTIRE PROCESS CAN BE AUTOMATED USING ROBOTS, BY REPLACING COMPONENTS AND MANUAL LABOUR
- THE RIGHT COMPONENTS CAN BE IDENTIFIED USING COMPUTER VISION
- A CLIENT FRIENDLY APP MAYBE DEVELOPED TO INCORPORATE ALL FEATURES TOGETHER SO THAT ALL THE INFORMATION CAN BE OBTAINED WITH A SINGLE CLICK
- BASED ON THE TELEMATICS DATA AN AI SOLUTION CAN BE MADE TO RECOMMEND THE COMPONENT THAT WILL PROVE BEST USE TO THE CUSTOMER GIVEN THE PREVIOUS RECORD OF THE EQUIPMENT IN USE. IN THIS WAY THE NUMBER OF REPLACEMENTS CAN BE MINIMIZED

## CONCLUSION

Thus an AI model is developed that will

- i) Predicts the date of an equipment failure
- ii) With the future scope of alerting the customer and checking the stock
- iii) With the future scope of recommending the best suitable replacement for the component.