NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALORE-575025

THE INSTITUTION OF ENGINEERS, NITK CHAPTER
Institutional Member ID: **IM0001491**
https://ie.nitk.ac.in

Extraction of Mechanical properties of Materials using Machine Learning

Problem statement:

Finding mechanical properties for complex alloys has always been a perplexing job. Major modern industries use costly, complex instruments like Instrumented-indentation to find mechanical properties for their materials.

We tried to tackle this issue by finding the Mechanical properties (that includes Proportional limit, Yield Point, Ultimate tensile strength, Fracture point) of complex Alloy like Al6061 at various temperatures (20°C, 100°C, 150°C, 200°C, 250°C and 300°C) using Machine Learning.

Proposed solution:

Using the concept of multiple regression lines (in Machine Learning) with a good accuracy, we want to find the optimum regression line out of multiple for our Proportional limit and similarly for other properties.

Technologies used:

Libraries - Numpy, Pandas, Seaborn, Matplotlib, Scikit-learn

Language - Python

Tool - Google Collab

Methodology:

We web searched for some complex alloy's stress-strain data, found for Al6061. The dataset is

stress-strain (Tensile and Microindentation) of Aluminium 6061 Alloy under uniaxial and plain strain tension at 6 different temperatures i.e. 20°C, 100°C, 150°C, 200°C, 250°C and 300°C.

After a lot of Data Cleaning, we loaded it to our DataFrame and did some Exploratory Analysis for finding some insights of our DataSet.

Plotted our DataSet using matplotlib, and then we used multiple linear regression lines and calculated their accuracy, and found the optimum line (one with the greater accuracy and has covered a certain number of data points from the dataset).

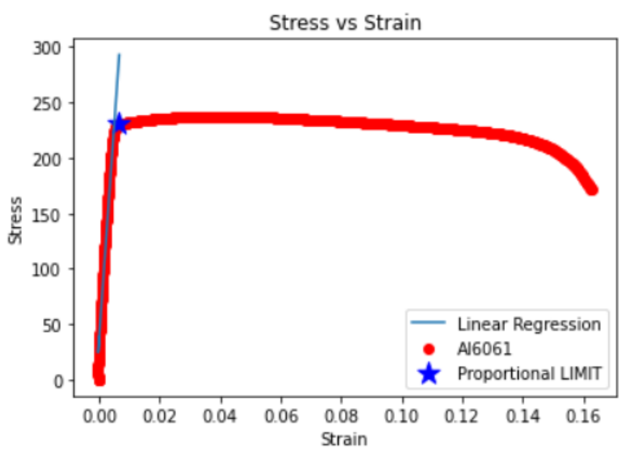
Here's the code for the same.

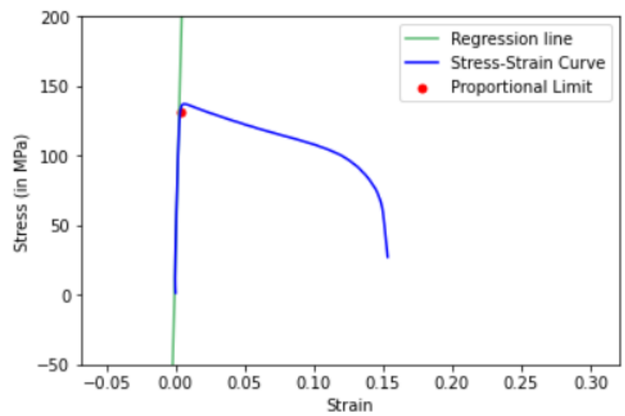
```
[ ] # Plotting multiple Regression lines and finding their accuracy
    import random
    prop_lim = random.randint(0,row)

    for i in range(2, row):
      #Change to DataFrame
      x = pd.DataFrame(df.loc[0:i , ['Strain']])
      y = pd.DataFrame(df.loc[0:i , ['Stress_MPa']])

      model = LinearRegression().fit(x, y)
      y_new = model.predict(x)

      acc = round(slm.r2_score(y, y_new), 2)
      if acc >= 0.95 :
        prop_lim = i
    prop_lim
```

Here's another way to do the same.

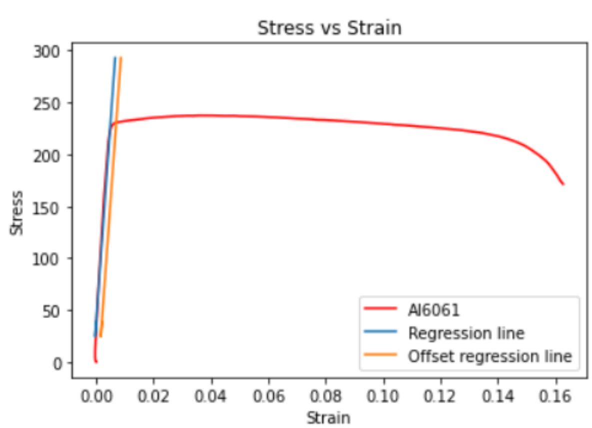
```
[ ] #finding multiple regression line and plotting the line with highest accuracy
    m = len(X)
    a = np.zeros(m)
    c = np.zeros(m)
    r2 = np.zeros(m)

    for i in range(50,m):
      mean_x = 0
      mean_y = 0
      for j in range(i):
        mean_x += X[j]
        mean_y += Y[j]
      mean_x = mean_x/i
      mean_y = mean_y/i
      num = 0
      den = 0
      for j in range(i):
        num += (X[j]-mean_x)*(Y[j]-mean_y)
        den += (X[j]-mean_x)**2
      a[i] = num/den
      c[i] = mean_y-(a[i]*mean_x)

      ss_d = np.zeros(m)
      ss_n = np.zeros(m)

      for j in range(i):
        predict_y = c[i]+a[i]*X[j]
        ss_d[i] += (Y[j]-mean_y)**2
        ss_n[i] += (Y[j]-predict_y)**2
      r2[i] = 1-(ss_n[i]/ss_d[i])

      i_max = np.argmax(r2)
      r2_final = np.max(r2)
      a_final = a[i_max]
      c_final = c[i_max]
    print(a_final,c_final,r2_final)
```

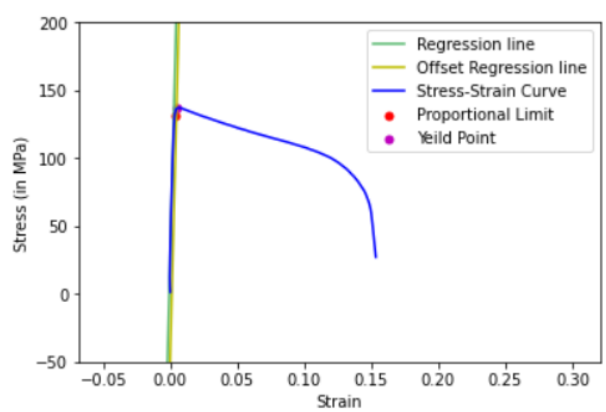
We took the optimum regression line and made our model to find the proportional limit by considering the intersecting point of the regression line and the stress-strain curve.



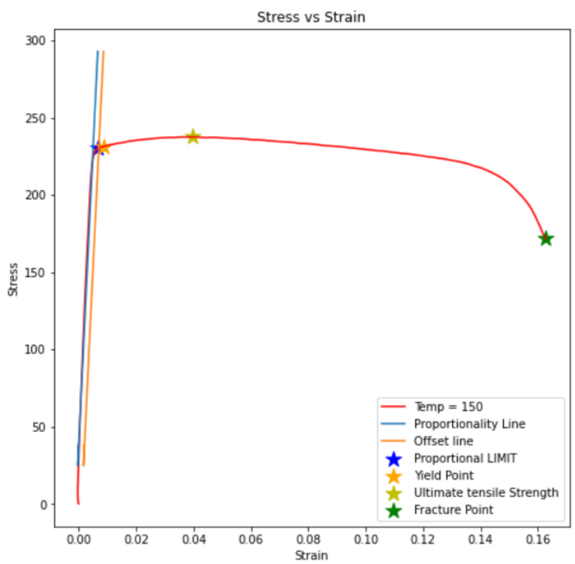


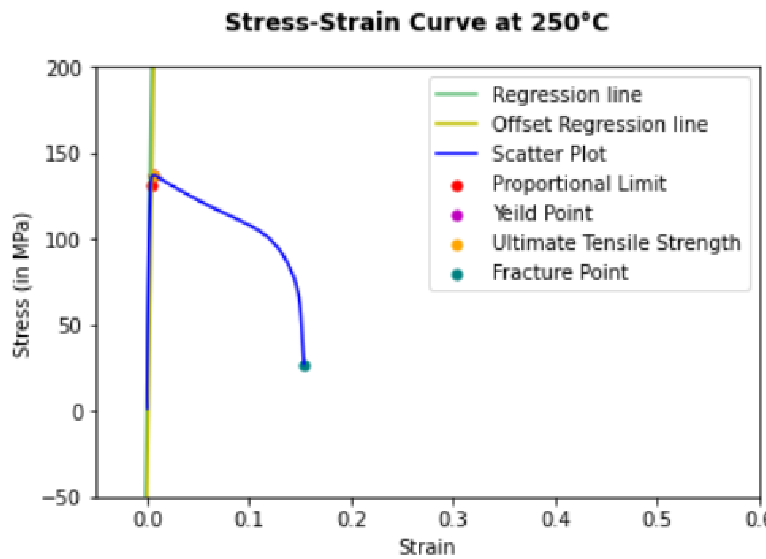
Then, we took an offset of 0.2% of the regression line for finding yield point in a similar way.





Highest stress was ultimate tensile strength and last possible stress before breaking was fracture point. Here's the **Concluded PLOT** for the same.





Results:

Alloy Name	Temperature	Proportional Limit	Model Accuracy
Al6061	20°C	255.467444 MPa	95.66 %
Al6061	100°C	247.359740 MPa	97.72 %
Al6061	150°C	230.334518 MPa	94.54 %
Al6061	200°C	187.024834 MPa	94.58 %
Al6061	250°C	131.542554 MPa	95.42 %
Al6061	300°C	81.2411923 MPa	79.97 %

All six of our models were successful in finding the mechanical properties with an **average accuracy of 92.98% highest** being **97.72%** and **lowest** being **79.97%**.

All our findings were **verified from wikipedia** and all lied in the required range, in the given temperatures.

The following contains the **code for all six temperatures** we developed over the course of the project: IE MechProp & Github Repository

Future work:

1. We can extend this project by finding properties for some more temperatures and then predicting the mechanical properties for any given temperature provided it lies within the range of 20°C to 300°C.

2. We can do this for some more Aluminium Alloys (with different compositions) and then compare their properties with Al6061's to obtain some relationship between them.

Key Learnings:

1. Limitations of Instrumented Indentation.
2. Application of multiple regression lines.
3. Stress-Strain curve and its properties.
4. Teamwork and coordination.

References:

1. https://www.pnas.org/content/117/13/7052
2. https://www.sciencedaily.com/releases/2020/03/200316152210.htm
3. https://www.w3schools.com/python/
4. https://towardsdatascience.com/coding-deep-learning-for-beginners-linear-regression-part-1-initialization-and-prediction-7a84070b01c8
5. https://www.analyticsvidhya.com/blog/2020/08/exploratory-data-analysiseda-from-scratch-in-python/

Team:-

Mentor:

Priyank Agarwal (171MT031)

Mentee:

Dhruv Jha (191ME128),

Rajveer Singh (191ME268),

Utkarsh Jha (191MT051).