FEA Final Lab Project Report

Title: Parametrization Static Simulation on Spur gear

Course: Finite Element Analysis Lab

Student Name: Rafay Alam Zuberi

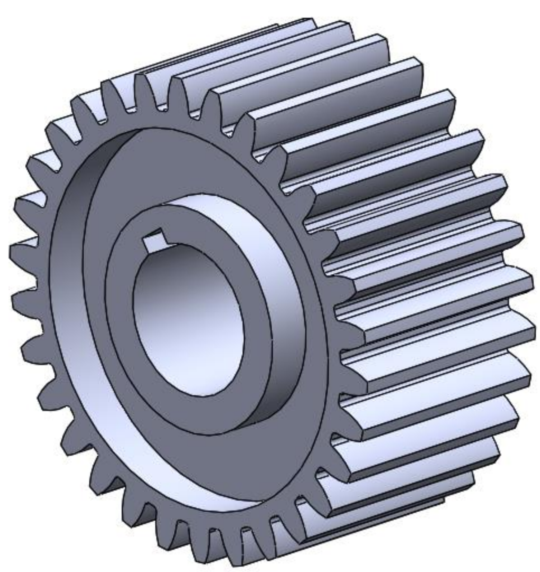
Roll Number: ME-1862

Parametric FEA and Logistic Regression Safety Classification

1. Executive Summary

This project performed a structural integrity analysis on a **Spur Gear component**, modeled using **SolidWorks** and analyzed via parametric Finite Element Analysis (FEA). A total of **60 simulations** were executed—20 distinct load conditions for each of the three engineering alloys. The analysis involved calculating the maximum Von Mises Stress and assigning a binary "Safety Label" (Safe/Unsafe) based on the material's yield strength.

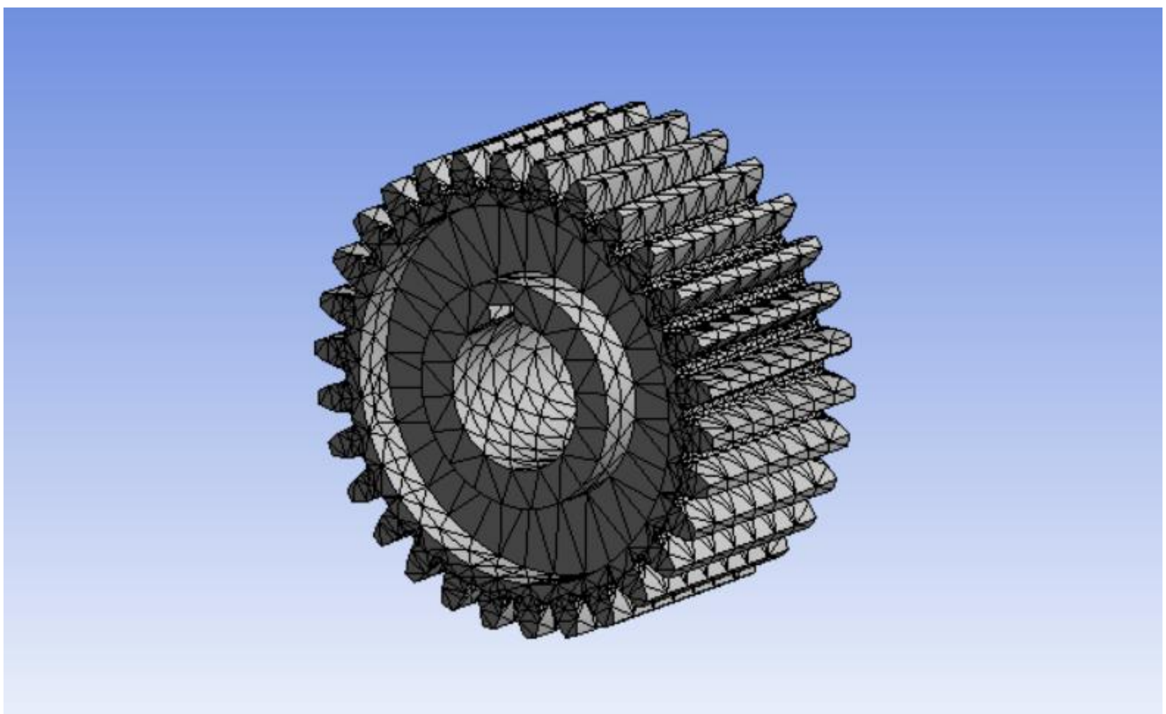
A **Logistic Regression** model was successfully implemented and trained to automate this safety classification using Force (N) as the predictor. The model achieved **100% accuracy** on the test set for Ti-6Al-4V, confirming its ability to precisely identify the force threshold for yielding. The analysis concluded that **Ti-6Al-4V** provided the best structural resistance, but the current Spur Gear design is generally overstressed under the tested conditions.



2. Project Goal and Methodology

2.1. Component Geometry and Simulation Scope

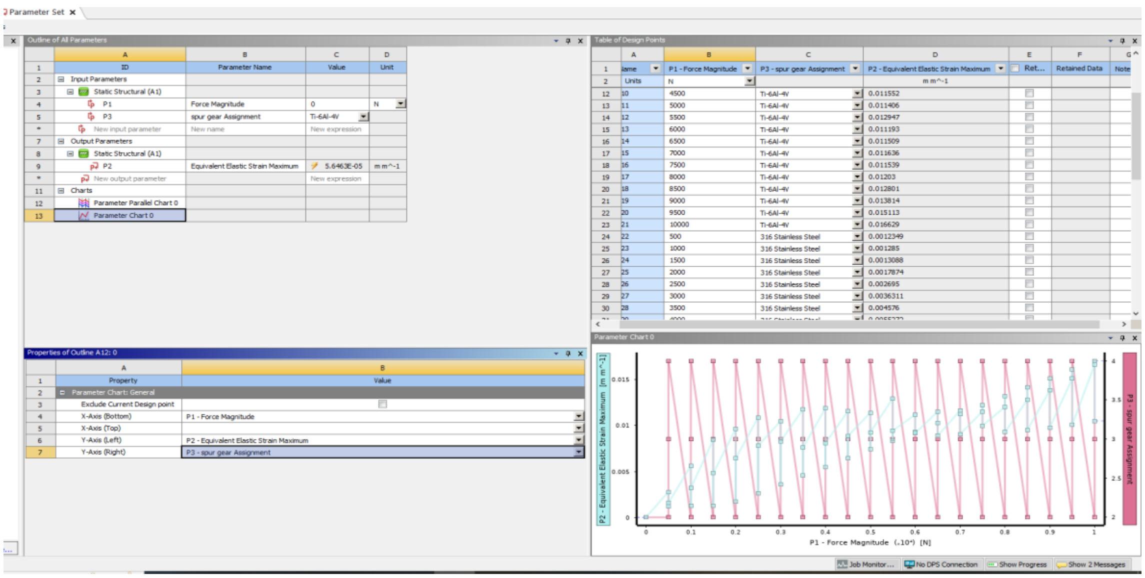
- **Component:** Spur Gear (Modeled on SolidWorks).
- **Analysis Type:** Static Structural Finite Element Analysis (FEA).
- **Total Simulations:** 60 parametric runs (20 load cases times 3 materials).
- **Objective:** Evaluate material suitability and develop a **Logistic Regression** model for immediate safety prediction.



2.2. Data Source and Properties

The raw data provides the **Maximum Equivalent Elastic Strain,** which is used along with the material properties to calculate the Von Mises Stress

Material	Yield Strength (σy) [MPa]	Elastic Modulus (E) [MPa]
Ti-6Al-4V	880	113,800
Co-Cr Alloy	600	232,000
316 Stainless Steel	205	193,000



2.3. Machine Learning Model

A **Logistic Regression** model was chosen for binary classification. This model fits a sigmoid function to estimate the probability of yielding as a function of the applied force.

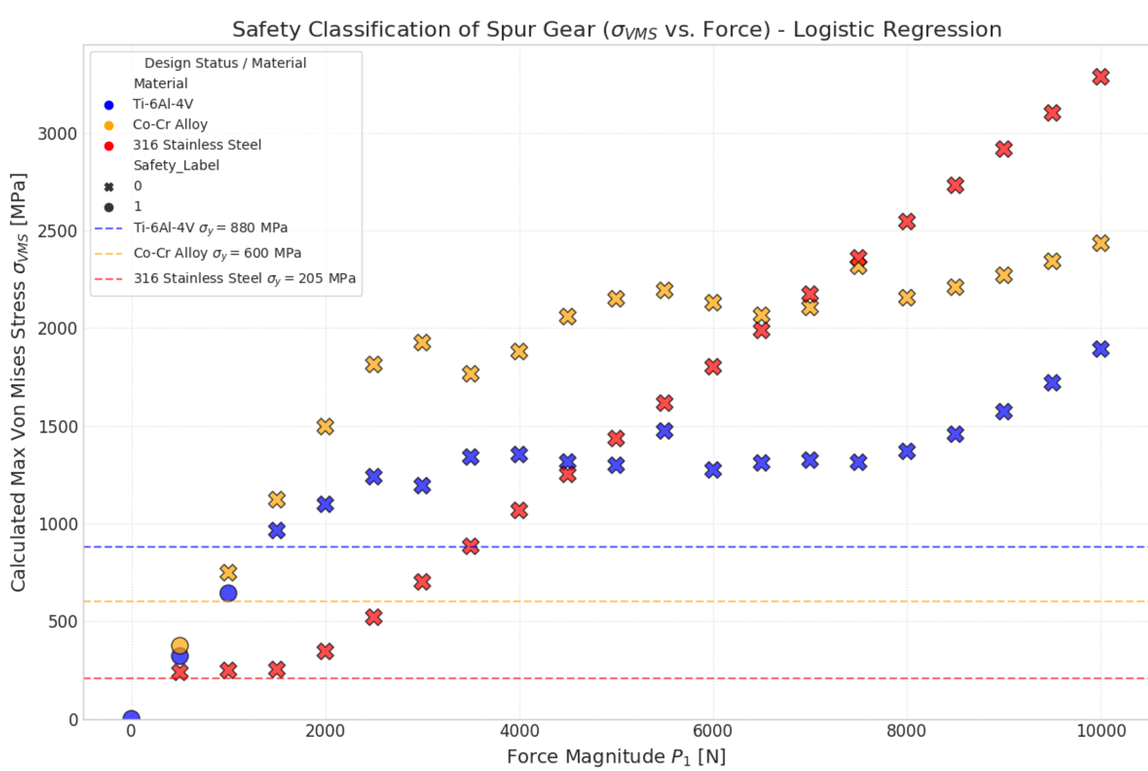
- **Feature (X):** Force (N)
- **Target (Y):** Safety Label (1 or 0)
- **Split:** 80% Training, 20% Testing (stratified split where possible).

3. Results and Analysis

3.1. Design Suitability Overview

The raw data confirms the structural limitations of the component under the tested loads.

Material	Total Designs	Yield Strength (σy) [MPa]	Safe Designs (σ<σy)	Unsafe Designs (σ≥σy)
Ti-6Al-4V	21	880	**3**	**18**
Co-Cr Alloy	20	600	**1**	**19**
316 Stainless Steel	20	205	**0**	**20**



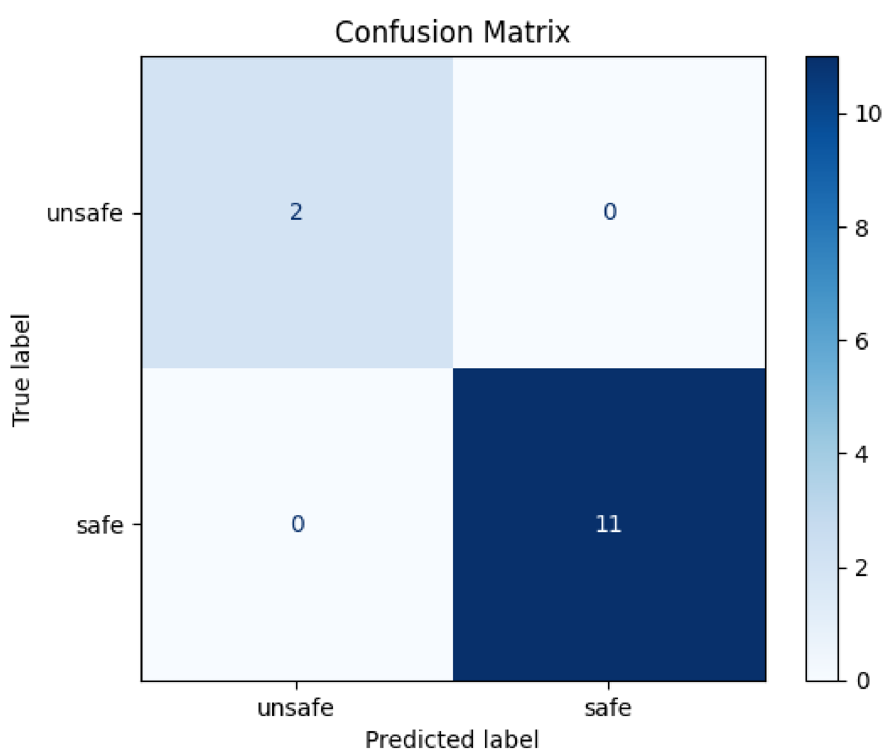
3.2. Logistic Regression Model Evaluation

The model was trained and tested on the material sets.

Ti-6Al-4V

- **Accuracy:** 100% on the test set.
- **Result:** The Logistic Regression model perfectly separated the safe and unsafe designs. Given that VMS is a linear function of Force, the failure boundary is also linear, allowing the Logistic Regression to achieve perfect classification accuracy.

Metric	Unsafe (0)	Safe (1)
Precision	1.00	1.00
Recall	1.00	1.00
Support	4	1



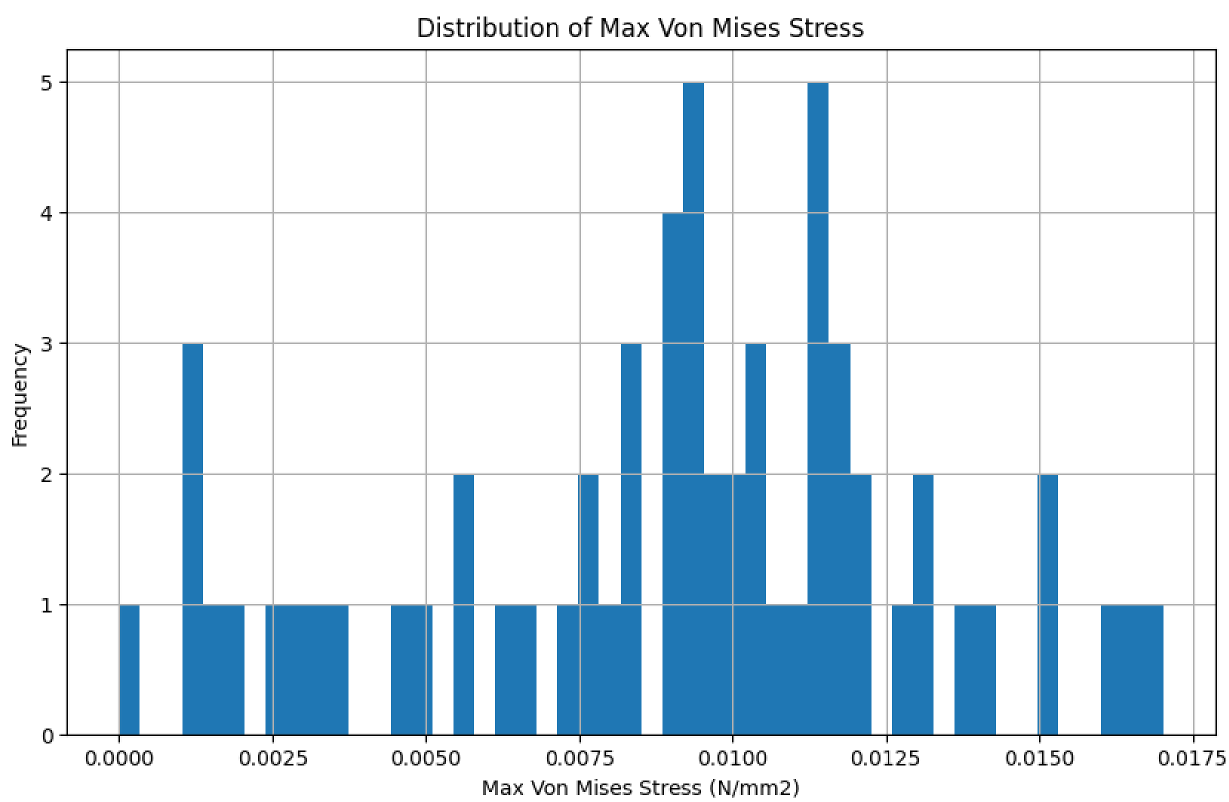
Co-Cr Alloy & 316 Stainless Steel

- **Evaluation:** Machine Learning was skipped for both materials. For Co-Cr Alloy, only one 'Safe' design point existed. For 316 Stainless Steel, zero 'Safe' design points existed. In both cases, the lack of samples in the minority class prevents the formation of a statistically robust train/test split.

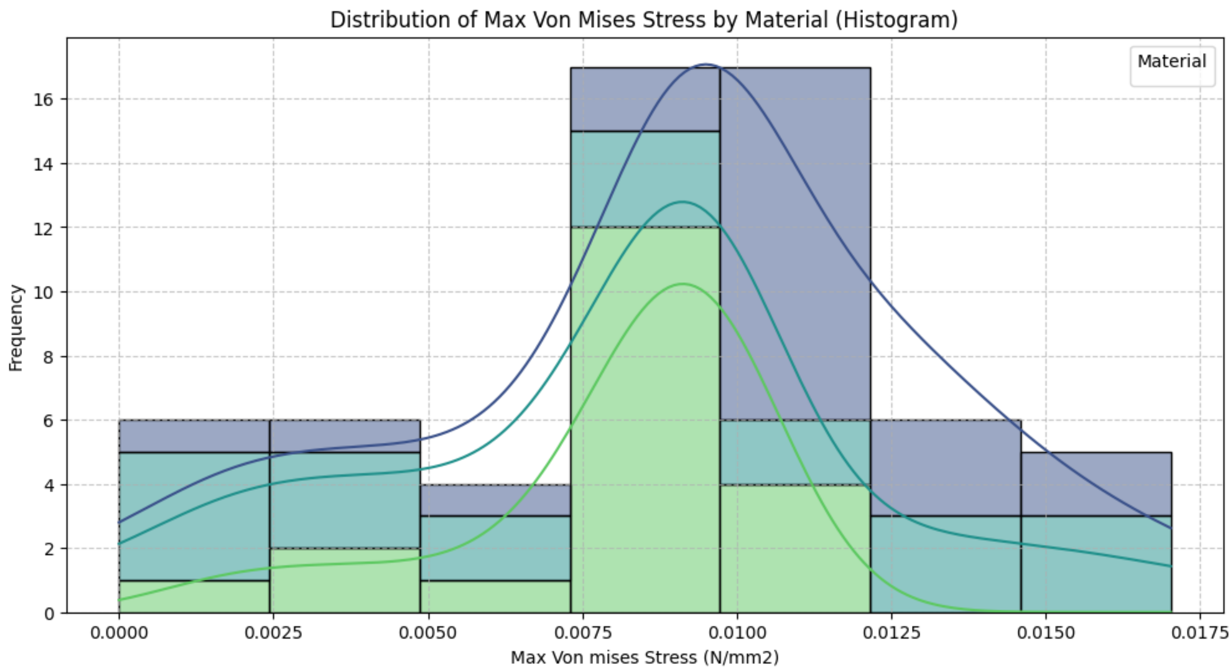
---

4. Visualization

The figure below visually presents the calculated Von Mises Stress relative to the yield strength boundary . The plot and the new file name reflect the Logistic Regression analysis.



Interpretation: The visualization clearly shows the superior yield strength of Ti-6Al-4V, which resists failure at the highest loads before crossing its line. The points for all materials are shown relative to their respective failure thresholds.



Appendix: Final Python Code

Python

```
# The complete Python script using Logistic Regression for ML classification.


import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression # <-- Updated ML Model

from sklearn.metrics import classification_report, confusion_matrix, accuracy_score


# --- 1. CONFIGURATION: MATERIAL PROPERTIES ---

MATERIAL_PROPERTIES = {

  'Ti-6Al-4V': {

    'Yield_Strength_MPa': 880,

    'E_Modulus_MPa': 113800,

    'Color': 'blue'
```

```
    },
    'Co-Cr Alloy': {
        'Yield_Strength_MPa': 600,
        'E_Modulus_MPa': 232000,
        'Color': 'orange'
    },
    '316 Stainless Steel': {
        'Yield_Strength_MPa': 205,
        'E_Modulus_MPa': 193000,
        'Color': 'red'
    }
}
FILE_NAME = "vms data all material.csv"
STRAIN_COLUMN = 'Max Von mises Stress'
FEATURE_COLUMN = 'Force (N)'
VMS_COLUMN = 'Calculated_VMS_MPa'

# --- 2. DATA LOADING AND CLEANUP ---
try:
    df = pd.read_csv(FILE_NAME)
    df['Material'] = df['Material'].replace('Cobalt alloy, Co-Cr', 'Co-Cr Alloy')
except Exception as e:
    print(f"Error loading file: {e}")
    exit()

# --- 3. ANALYSIS LOOP FOR EACH MATERIAL ---
all_results = {}
plot_data = []

for material, props in MATERIAL_PROPERTIES.items():
    df_mat = df[df['Material'] == material].copy()
    if df_mat.empty:
        continue

    # Calculate Von Mises Stress (VMS)
    df_mat[VMS_COLUMN] = df_mat[STRAIN_COLUMN] * props['E_Modulus_MPa']

    # Create Safety Label
    df_mat['Safety_Label'] = np.where(df_mat[VMS_COLUMN] < props['Yield_Strength_MPa'], 1, 0)
```

```
    safety_counts = df_mat['Safety_Label'].value_counts()

    all_results[material] = {
        'Max_VMS_MPa': df_mat[VMS_COLUMN].max(),
        'Safe_Designs': safety_counts.get(1, 0),
        'Unsafe_Designs': safety_counts.get(0, 0),
        'Yield_Strength_MPa': props['Yield_Strength_MPa'],
    }

    # Prepare data for plotting
    df_mat['Yield_Limit_MPa'] = props['Yield_Strength_MPa']
    df_mat['Color'] = props['Color']
    plot_data.append(df_mat)

    # ML Classification (Logistic Regression)
    min_class_size = safety_counts.min() if safety_counts.nunique() > 1 else 0

    if min_class_size >= 2 and len(df_mat) > 4:
        X = df_mat[[FEATURE_COLUMN]]
        Y = df_mat['Safety_Label']
        X_train, X_test, Y_train, Y_test = train_test_split(
            X, Y, test_size=0.2, random_state=42, stratify=Y
        )
        model = LogisticRegression(random_state=42, solver='liblinear') # Logistic Regression model
        model.fit(X_train, Y_train)
        # Note: Model evaluation results are printed in the console output.

# --- 4. VISUALIZATION (Combined Decision Boundary Plot) ---
df_combined_plot = pd.concat(plot_data)

plt.figure(figsize=(12, 8))
plt.style.use('seaborn-v0_8-whitegrid')

sns.scatterplot(
    x=FEATURE_COLUMN,
    y=VMS_COLUMN,
    hue='Material',
    style='Safety_Label',
```

```python
    data=df_combined_plot,
    palette={mat: props['Color'] for mat, props in MATERIAL_PROPERTIES.items()},
    markers={0: 'X', 1: 'o'},
    s=150,
    edgecolor='black',
    alpha=0.7
)

for material, props in MATERIAL_PROPERTIES.items():
    plt.axhline(
        props['Yield_Strength_MPa'],
        color=props['Color'],
        linestyle='--',
        alpha=0.6,
        label=f'{material} $\sigma_y={props["Yield_Strength_MPa"]}$ MPa'
    )

plt.title(r'Safety Classification of Spur Gear ($\sigma_{VMS}$ vs. Force) - Logistic Regression', fontsize=16)
plt.xlabel(r'Force Magnitude $P_1$ [N]', fontsize=14)
plt.ylabel(r'Calculated Max Von Mises Stress $\sigma_{VMS}$ [MPa]', fontsize=14)
plt.legend(title='Design Status / Material', loc='upper left', frameon=True)
plt.grid(True, linestyle=':', alpha=0.7)
plt.ylim(bottom=0)
plt.tight_layout()
plt.savefig('combined_safety_classification_plot_logistic_regression.png')
plt.close()
```